Universidade de Aveiro
2009
Departamento de Electrónica, Telecomunicações e Informática

Marco Alexandre de Oliveira Matias

RTS-sec: Privacidade e Segurança em Redes Telemáticas para a Saúde

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática
2009

Marco Alexandre de Oliveira Matias

RTS-sec: Privacidade e Segurança em Redes Telemáticas para a Saúde

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Professor Doutor André Zúquete, Professor Auxiliar no Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e do Professor Doutor João Paulo Cunha, Professor Associado no Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri

presidente

Prof. Dr. José Luís Guimarães Oliveira
professor associado da Universidade de Aveiro

Prof. Dr. Carlos Nuno da Cruz Ribeiro
professor auxiliar do Instituto Superior Técnico da Universidade Técnica de Lisboa

Prof. Dr. André Ventura Marnoto Zúquete
professor auxiliar da Universidade de Aveiro

Prof. Dr. João Paulo Trigueiros da Silva Cunha
professor associado da Universidade de Aveiro

agradecimentos

Aos orientadores, especialmente ao professor André Zúquete, pelo apoio, fornecimento de material para trabalhar com o Cartão de Cidadão, e documentos fornecidos.
Ao Hélder Gomes, que fez o trabalho principal sobre a autenticação na RTS.
Aos meus pais, que sempre me apoiaram.
Aos meus amigos, que me incentivaram e debateram comigo algumas ideias.

Muito Obrigado!

palavras-chave

Segurança informática, privacidade de dados, redes telemáticas de serviços, sistemas de informação na Saúde.

resumo

Este trabalho apresenta o estudo de uma solução que permite a autenticação de profissionais na Rede Telemática da Saúde (RTS) com o Cartão de Cidadão. São definidas ligeiras alterações numa arquitectura anteriormente definida de forma a adaptar o mecanismo de autenticação ao uso do Cartão de Cidadão. Uma Infra-Estrutura de Chave Pública (PKI) serve de base para a solução apresentada. É descrito como se gera um certificado de chave pública que aproveita o par de chaves de autenticação existente no Cartão de Cidadão.

keywords

Informatic security, data privacy, telematic service networks, health information systems.

abstract

This work presents a study of a solution which allows the authentication of health professionals to access a Telematic Health Information System (RTS – Rede Telemática de Saúde) using the Citizen's Card (Cartão de Cidadão). Some changes in a previous defined architecture are defined, in a way to adapt the authentication mechanism so it can use the Citizen's Card. A Public Key Infrastructure provides the basis for the presented solution. It's described how to generate a public key certificate that can reuse the authentication key pair existing in the Citizen's Card.

Índice

1 Introdução **1**

1.1 Motivação 1

1.2 Problema 2

1.3 Objectivos 2

1.4 Proposta de Solução 4

2 Contexto **5**

2.1 Rede Telemática de Saúde 5

2.1.1 Arquitectura 6

2.2 Conceitos Básicos 7

2.2.1 Assinaturas Digitais 7

2.2.2 Criptografia de Chave Pública 7

2.2.3 Infraestrutura de Chave Pública e Certificados Digitais . . 8

2.2.4 Entidades Certificadoras 9

2.2.5 Cadeias de Certificação e Verificação 13

2.2.6 Chaveiros Protegidos e Ficheiros Certificados 14

2.2.7 Revogação de Certificados 16

2.2.8 Autenticação Mútua (Cliente e Servidor) 17

2.3 PKCS#11 17

2.4 Cartão de Cidadão 18

2.4.1 Como é Constituido o Cartão de Cidadão 18
2.4.2 Autenticação Simples com o Cartão de Cidadão 20
2.4.3 Middleware do Cartão de Cidadão 22
2.4.4 API eID Lib 25
2.4.5 Normas dos Leitores de Cartões 25
2.4.6 Cartão de Cidadão e *Smart Card* Genérico 26
2.5 O Projecto OpenSSL 27

3 Trabalhos Relacionados **29**
3.1 Alemanha - Electronic Health Card (eHC) e Health Professional Card (HPC) 29
3.2 Uso de *Smart Cards* em Redes Telemáticas de Saúde 31
3.3 Arquitectura Definida pelo H. Gomes 31
3.3.1 Introdução 32
3.3.2 Descrição da Arquitectura 32
3.3.3 Adaptaçao da Arquitectura de Autenticação para o Uso do Cartão de Cidadão 40

4 Estudo da Solução com Código Activo **43**
4.1 Modo de Funcionamento das Credenciais 46
4.1.1 Protocolos 47
4.1.2 Ligações (Bindings) 48
4.1.3 Perfis 48
4.1.4 SAML e a RTS 49
4.2 Tecnologias Abordadas 49
4.2.1 JavaScript 49
4.2.2 Java Applets 50
4.2.3 ActiveX 52

4.2.4 Macromedia Flash 52
4.3 Avaliação 52

5 Estudo da Solução sem Código Activo 53
5.1 Wrapping do Cartão de Cidadão 53
5.1.1 Modo de Funcionamento 54
5.1.2 Descrição dos Componentes Utilizados 55
5.1.3 O Módulo PKCS#11 56
5.1.4 Interacção entre o *browser* Firefox e a PKCS#11 57
5.2 Exploração de Tokens via PKCS#11 com o Firefox 58

6 Arquitectura Proposta 65
6.1 Apresentação 65
6.2 Certificados Digitais 66
6.2.1 Entidades com Certificados 66
6.2.2 Tipos de Certificados 67
6.2.3 Formato dos Certificados 67
6.2.4 Gestão de Certificados 68
6.2.5 Emissão/Renovação 68
6.2.6 Armazenamento 68
6.2.7 Cadeias de Certificação 69
6.3 Geração de um Pedido de Certificado 69
6.4 Processo de Autenticação 71

7 Avaliação 73
7.1 Arquitectura Avaliada 73

8 Conclusões 75

Lista de Figuras

2.1 Arquitectura da RTS [1] 6
2.2 SSL - Autenticação Mútua 12
2.3 Cartão de Cidadão - Frente 19
2.4 Cartão de Cidadão - Verso 19
2.5 Informação e Aplicações Residentes no Chip 20
2.6 Autenticação Genérica com o Cartão de Cidadão 21
2.7 Interacção entre aplicações e o módulo PKCS#11 22

3.1 Cartão de Saúde do Profissional (HPC) 30
3.2 Cartão de Saúde do Utente (eHC) 31
3.3 Visão Geral da Arquitectura da RTS 33
3.4 Modelo de Comunicação e Autenticação 35
3.5 Modelo de Confiança Entre a RTS e as IS 37
3.6 Arquitectura Interna das EC 38
3.7 Construção de Cadeia de Confiança 39
3.8 Comunicação Segura Entre o Profissional e a RTS 40

4.1 Solução com Código Activo 44
4.2 Relação entre a RTS e uma IS 45
4.3 Modelo de autenticação SAML 47

5.1 Arquitectura 54
5.2 Interacção Entre o Browser e o Cartão de Cidadão usando o Wrapper 55
5.3 Caixa de diálogo de inserção de PIN 57
5.4 Adicionar um Módulo - Parte 1 59
5.5 Adicionar um Módulo - Parte 2 59
5.6 Adicionar um Módulo - Parte 3 60
5.7 Gestor de Certificados do Firefox 62
5.8 Selecção de Certificado 63

6.1 Cadeias de Certificação: IS - RTS 70

Lista de Tabelas

2.1 Objectos Existentes no Cartão de Cidadão 25

5.1 Alguns Atributos da Cryptoki 58

5.2 Atributos de um Certificado 61

Capítulo 1

Introdução

A Rede Telemática da Saúde (RTS)[1] implementa uma infra-estrutura de TIC para agilizar a comunicação clínica na região de Aveiro. Desenvolvida no âmbito do projecto RTS homónimo, a rede coloca novas possibilidades de acesso integrado a Sistemas de Informação existentes e de comunicação electrónica entre profissionais de saúde.

A RTS lança bases para quebrar as barreiras de isolamento entre as fontes de conhecimento clínico, disponíveis na região e pertinentes para a continuidade de cuidados. Utilizando a RTS, por exemplo, um médico pode, através de um "Portal Regional de Saúde", aceder do seu posto de trabalho (e.g.: Centro de Saúde em Sever do Vouga) à informação clínica do seu doente gerada em outras instituições (e.g.: consulta no Hospital Infante D. Pedro, Aveiro).

1.1 Motivação

Anteriormente foi proposta pelo H. Gomes [2] uma arquitectura de autenticação para a RTS com recurso ao uso de *smart cards*. Essa arquitectura resolve o problema de autenticação, mas acarreta alguns custos para as Instituições de Saúde, em cartões para os seus Profissionais, e mesmo para os Profissionais, pois é mais um cartão que têm que trazer consigo (não faz muito sentido transportar dois cartões capazes de efectuar as mesmas funcionalidades quando apenas um é o suficiente).

Esses custos podem ser minimizados efectuando um aproveitamento dos re-

[1] http://www.rtsaude.pt/

cursos existentes no Cartão de Cidadão [2], sendo isso a base da motivação para este estudo.

A substituição de *smart cards* pelo Cartão de Cidadão traz algumas vantagens e desvantagens. Uma vantagem é que, por ser obrigatório, torna-se mais barato para uma organização aproveitar alguns recursos nele existentes. Além disso, outra vantagem do Cartão de Cidadão é a sua multifuncionalidade, permitindo que o seu uso possa ter vários fins, tudo no mesmo cartão.

Torna-se incómodo transportar varios documentos, cada um com a sua funcionalidade. Daí ter surgido a ideia de adicionar a funcionalidade de autenticação na RTS ao Cartão de Cidadão. A principal desvantagem é que a introdução do Cartão de Cidadão na arquitectura existente não é trivial, e implica algumas alterações que vão ser mencionadas ao longo do texto.

Aproveitando o modelo de PKI definido anteriormente pelo H. Gomes, são estudadas diversas soluções para a autenticação na RTS com o uso do Cartão do Cidadão.

1.2 Problema

A criação de uma infraestrutura que permite a autenticação em sistemas telemáticos biomédicos foi anteriormente abordada pelo H. Gomes [2]. O novo problema que surge é a adaptação dessa infraestrutura para que possa ser usada com o Cartão de Cidadão, em vez de *smart cards* comuns, ficando restrita às limitações impostas pelo Cartão de Cidadão.

A arquitectura do H. Gomes assume que um único *smart card* é o suficiente para autenticar o profissional perante a sua instituição e a RTS. Isso não é possível com o Cartão de Cidadão porque o mesmo não permite que lhe sejam introduzidas novas credenciais. Portanto a arquitectura do H. Gomes, à partida, não permite a utilização directa, simples, do Cartão de Cidadão.

1.3 Objectivos

O objectivo deste trabalho consistiu em estudar formas alternativas de adaptar o Cartão de Cidadão à arquitectura definida pelo H. Gomes.

[2] http://www.cartaodecidadao.pt/

Apesar do Cartão de Cidadão permitir a autenticação de um determinado cidadão, ele não possui nenhuma funcionalidade nativa de definir de maneira segura o cargo desempenhado por esse cidadão numa Instituição de Saúde.

O primeiro objectivo consistiu em efectuar uma análise dos recursos disponíveis pelo Cartão de Cidadão e definir quais poderiam ser aproveitados.

O segundo objectivo consistiu na definição de um mecanismo para que a autenticação pudesse ser efectuada com base no cargo desempenhado por um determinado Profissional numa Instituição de Saúde.

Para isso é necessário definir credenciais com a informação sobre o cargo do Profissional e a Instituição de Saúde a que pertence. Essas credenciais devem estar relacionadas apenas com o cartão do Profissional que tem permissão para as usar e serem inválidas com qualquer outro. Isto não deve impedir que um cartão tenha várias credenciais associadas a ele. Se for possível, é desejada a inclusão das credenciais no cartão ao qual estão relacionadas.

O terceiro objectivo seria a utilização das credenciais definidas nos objectivos anteriores por um *browser*.

Uma alternativa consistiu em procurar soluções que usassem código activo que corre no *browser*, manipulando directamente o Cartão de Cidadão e as credenciais a ele associadas.

A outra alternativa consistiu numa solução que se serve do comportamento nativo do *browser* face ao protocolo SSL e que manipula o Cartão de Cidadão via PKCS#11.

Por enquanto a autenticação apenas funciona com o *browser* Firefox, que usa a interface PKCS#11. O Internet Explorer não usa directamente a interface PKCS#11, usa a interface Crypto API/CSP [3].

Apenas foi explorado o uso do *browser* Firefox, por ser um *browser* suportado por vários sistemas operativos, permitindo aos profissionais que efectuem o processo de autenticação na RTS quer usem ambiente Windows ou Linux. Outra razão pela qual o teste foi efectuado com este *browser* é a sua funcionalidade de adicionar módulos, podendo ser escolhido o módulo do *wrapper* sempre que se quer efectuar um teste com um módulo diferente. O Internet Explorer não possui essa opção, no entanto existe a possibilidade de adicionar o *wrapper* como biblioteca PKCS#11 usada pelo CSP, mas essa opção não chegou a ser explorada.

O *wrapper* foi implementado conforme os requisitos do *browser* Firefox, e a sua adaptação ao Internet Explorer ainda é uma área por explorar, mas que não deve trazer dificuldades. Neste documento são feitas várias referências ao

browser, que em certos casos devem ser entendidas como *browser* Firefox. Assim basta introduzir o Cartão de Cidadão num leitor e aceder normalmente ao portal da RTS com o *browser* Firefox para que seja feita a autenticação.

1.4 Proposta de Solução

Neste trabalho são estudadas algumas soluções possíveis. Uma solução implica código activo no *browser* usando uma *applet* Java que trata da verificação e validação da associação entre as credenciais de um determinado Profissional e o Cartão de Cidadão pertencente a ele. Outra função da *applet* é o estabelecimento da ligação entre a máquina cliente usada pelo Profissional e o servidor da RTS. A outra solução tem o mesmo objectivo mas usa apenas as funcionalidades nativas do *browser* servindo-se da biblioteca PKCS#11 fornecida com o Cartão de Cidadão, com algumas funcionalidades adicionadas. Basicamente, consiste no uso de um *wrapper* para biblioteca PKCS#11.

A solução escolhida foi a mais promissora, que não implica código activo, apenas é preciso instalar um módulo, além do *middleware* do Cartão do Cidadão. Esse módulo é designado por *wrapper*, que é carregado pelo *browser* permitindo que seja importado outro certificado além dos que vêm no cartão, através da interface criptográfica PKCS#11 [4] possibilitando uma ligação segura entre o profissional e o servidor da RTS.

Trata-se da solução mais simples e fácil de implementar do que uma solução de código activo, pois não requer código activo nem do lado do cliente nem do lado do servidor. Traz menos carga de processamento aos servidores, que não necessitam de correr código extra, e a integração com o modelo definido pelo H. Gomes é quase directa, porque as credenciais já estão no formato de um certificado X.509 e os métodos para extrair as credenciais do certificado são os mesmos.

O *wrapper* foi implementado de raiz, aproveitando as funcionalidades da *pteidlib.dll*, que permite fazer o acesso ao bloco de notas do Cartão de Cidadão, aproveitando também funcionalidades do módulo da PKCS#11 fornecido no *middleware* do Cartão de Cidadão e algumas funções da biblioteca de funções do OpenSSL que permitem manipular estruturas de dados de certificados X.509v3, facilitando a sua conversão para os modelos de estruturas pedidos pela norma PKCS#11.

O método de geração de certificados X.509v3 foi implementado sobre o código do OpenSSL, adicionando a opção de criação de certificados para o Cartão de Cidadão.

Capítulo 2

Contexto

Este capítulo descreve o que é a Rede Telemática de Saúde. Também é feito um apanhado de algumas noções básicas de termos que vão ser referidos em capítulos seguintes e um resumo da arquitectura existente. Além disso é feita uma descrição do Cartão de Cidadão, e das APIs utilizadas.

2.1 Rede Telemática de Saúde

A RTS (Rede Telemática de Saúde) [1] é um projecto desenvolvido pela Universidade de Aveiro com o objectivo de permitir a consulta de um conjunto de informações clínicas fornecidas por um grupo de Instituições de Saúde aderentes. Cada instituição usa o seu próprio sistema para produzir dados clínicos, que podem ser procurados e apresentados de várias maneiras pela RTS. O seu objectivo não é substituir os sistemas existentes, mas sim fornecer uma vista global dos dados clínicos de um paciente independentemente da Instituição de Saúde que guarda os seus dados.

A RTS fornece dois tipos de portais para aceder a registos electrónicos. São eles o Portal do Profissional e o Portal do Utente. O Portal do Profissional consiste num servidor Web acedido através da RIS (Rede Informática de Saúde), que é uma rede privada a nível nacional que faz a ligação entre as várias Instituições de Saúde inclusive as que fazem parte da RTS.

Os profissionais têm assim acesso à informação clinica fornecida pela RTS usando um *browser* Web de um computador ligado à RIS. O Portal do Utente tem o objectivo de permitir aos utentes um meio de comunicação entre eles e o seu

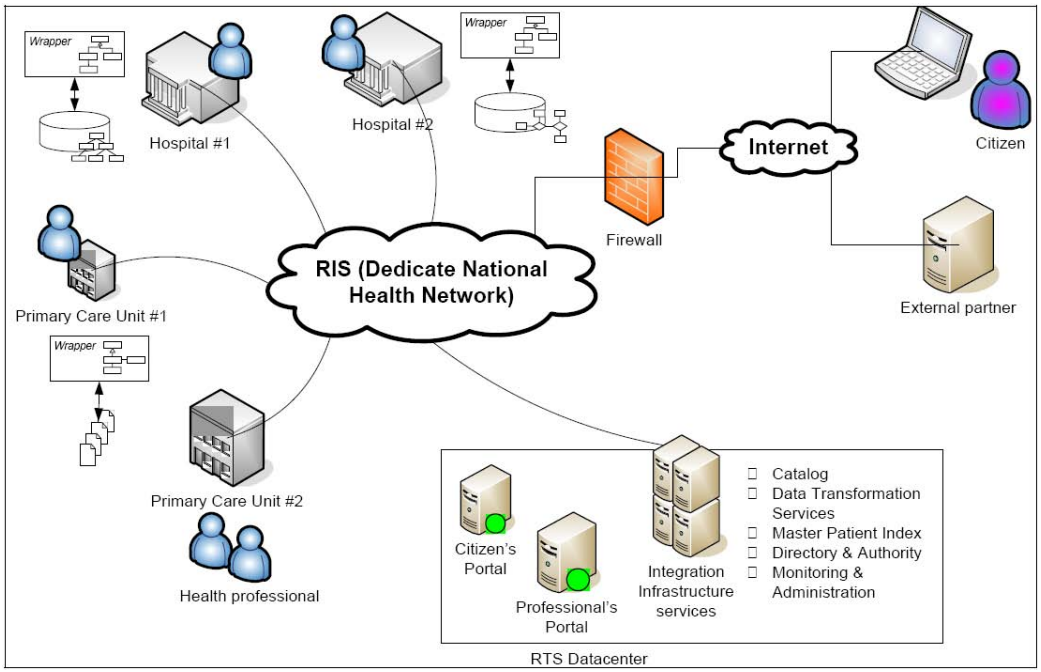

Figura 2.1: Arquitectura da RTS [1]

médico de família e de gerir problemas de saúde como por exemplo a renovação de receitas ou marcação de consultas [5] [6].

2.1.1 Arquitectura

A arquitectura da RTS é apresentada na Figura 2.1, onde se podem ver Instituições de Saúde (IS) com os respectivos Profissionais, Utentes, outras Entidades Externas e a base de dados da RTS.

Os dados clínicos de pacientes são produzidos e armazenados nas Instituições de Saúde. Para permitir a partilha e integração da informação médica, cada Instituição de Saúde possui um *Wrapper*, que faz a interligação entre os sistemas de informação da Instituição de Saúde respectiva e o *RTS Data Center*.

Os Portais (do Utente e do Profissional) são aplicações Web, vistas pelos utilizadores como servidores HTTP aos quais acedem através de um *browser* comum. A comunicação interna entre Portais e *Wrappers* é baseada em *Web Services*, utilizando-se objectos SOAP.

2.2 Conceitos Básicos

2.2.1 Assinaturas Digitais

Quando acontece uma transferência electrónica de documentos importantes, normalmente é necessário certificar de uma maneira fiável quem é na verdade o remetente (autor) de um determinado documento. Uma abordagem para certificar a origem de documentos e ficheiros é através do uso de uma assinatura digital. A assinatura digital de documentos usa a criptografia de chave pública como base matemática.

2.2.2 Criptografia de Chave Pública

A criptografia de chave pública é uma ciência matemática usada para fornecer confidencialidade e autenticidade na troca de informação através do uso de algoritmos criptográficos que funcionam com chaves públicas e privadas. Estes algoritmos criptográficos são usados para assinar documentos digitalmente, verificar a assinatura digital, e cifra e decifra de documentos.

As chaves públicas e privadas são um par de chaves criptográficas matematicamente relacionadas. A cada chave pública corresponde exactamente uma chave privada e vice-versa. Para usar a criptografia de chave pública é preciso ter uma chave pública e a respectiva chave privada.

A Chave Pública é um número (sequência de bits), que é normalmente atribuída a uma pessoa. Uma chave pública pode ser usada para verificar assinaturas digitais, criadas com a correspondente chave privada, tal como cifrar documentos que só podem depois ser decifrados pelo dono da respectiva chave privada.

As chaves públicas não são segredo para ninguém e normalmente estão publicamente disponíveis. A chave pública de uma determinada pessoa A deve ser conhecida por qualquer outra pessoa B que queira comunicar com a pessoa A usando a criptografia de chaves públicas.

A Chave Privada é um número (sequência de bits), conhecido apenas pelo seu dono. Com a sua chave privada, uma pessoa pode assinar documentos e decifrar documentos que foram cifrados com a respectiva chave pública. De certa maneira, as chaves privadas assemelham-se a palavras-chave de acesso bem conhecidas, que são um método muito utilizado de autenticação na Internet. A semelhança é que com a chave privada, tal como com a palavra-chave, uma pessoa pode provar

a sua identidade, ou seja, autenticar-se. Adicionalmente, como com as palavras-chave, as chaves privadas são supostamente secretas para todos excepto para o seu dono.

Ao contrário das palavras-chave de acesso, as chaves privadas não são assim tão pequenas que possam ser decoradas, e assim o seu local de armazenamento necessita de cuidados especiais. Se uma chave privada cai nas mãos de outra pessoa, por exemplo, se for roubada, a comunicação baseada na criptografia de chave pública dependendo desta chave privada deixa de ter significado. Em casos como este, a chave comprometida deve ser anunciada como inválida e ser substituída para que seja possível comunicar novamente de forma segura com o dono da chave.

Para este efeito, a criptografia de chave pública usa tais algoritmos criptográficos que é praticamente impossível para a matemática actual e para os computadores actuais encontrarem a chave privada de uma pessoa, sabendo a sua chave pública. De facto, a descoberta de uma chave privada correspondente a uma chave pública é possível na teoria, mas o tempo e o poder computacional necessário tornam tais operações insignificativas.

De um ponto de vista matemático é impossível assinar um documento sem saber a sua chave privada da pessoa que a assina. Também é impossível decifrar um documento que foi cifrado com a chave pública de uma certa pessoa sem saber a respectiva chave privada.

A assinatura digital permite certificar a origem e a integridade de informação transmitida electronicamente. Durante o processo de assinatura digital, é adicionada informação ao documento (a assinatura digital), que é calculada baseada no conteúdo do documento e uma chave privada. Numa fase final, esta informação pode ser usada para verificar a origem do documento assinado.

A assinatura digital é um número (sequência de bits), calculado matematicamente quando é assinado um determinado documento (mensagem). Este número depende do conteúdo da mensagem, do algoritmo usado para assinar e da chave privada usada para executar a assinatura. A assinatura digital permite que o receptor verifique qual é a origem da informação e a sua integridade.

2.2.3 Infraestrutura de Chave Pública e Certificados Digitais

A Infraestrutura de Chave Pública (PKI) fornece a arquitectura, técnicas de organização, práticas, e procedimentos que suportam, através de certificados digitais, a aplicação da criptografia de chave pública com a finalidade de assegurar a troca de informação

sobre redes inseguras. Para a emissão e controlo desses certificados digitais, a PKI depende das Entidades Certificadoras, que permitem a confiança entre diferentes organizações, que participam na comunicação segura baseada em chaves públicas e privadas.

Os certificados digitais associam uma certa chave pública a uma pessoa. Eles são emitidos por uma autoridade especial (Entidade Certificadora, *Certification Authority* - CA) com precauções de segurança estritas, que garante autenticidade deles. Pode-se pensar num certificado digital como um documento electrónico, que certifica que uma chave pública pertence a uma certa pessoa. Na prática, para efeitos de assinatura digital, os mais usados são os certificados X.509.

X.509 é uma norma generalizada para certificados digitais. Um certificado digital X.509 contem a chave pública de uma determinada pessoa, dados privados sobre esta pessoa (nome, organização, etc.), informação sobre a entidade certificadora que emitiu o certificado, informação sobre o período de validade, informação sobre os algoritmos criptográficos usados e outros detalhes variados.

2.2.4 Entidades Certificadoras

A entidade certificadora (CA) é uma instituição intitulada para emitir certificados digitais e para os assinar com a sua própria chave privada. A finalidade dos certificados é confirmar que uma dada chave pública pertence a uma dada pessoa, e a finalidade das entidades certificadoras é de confirmar que um dado certificado é válido e de confiança. Neste sentido, as entidades certificadoras são entidades terceiras e imparciais que fornecem segurança de alto nível na troca de informação baseada em sistemas computacionais.

Se uma entidade certificadora emitiu um certificado digital a uma determinada pessoa e assinou que este certificado pertence realmente à pessoa em causa, pode-se acreditar que a chave pública do certificado pertence de facto à pessoa, caso se confie na entidade certificadora.

Dependendo do nível de segurança necessário, podem-se usar certificados com diferentes níveis de confiança. Para a emissão de alguns tipos de certificados, só é necessário o endereço de e-mail do seu dono, enquanto para a emissão de outros é necessária a presença pessoal do seu dono, que por exemplo assina um documento em papel num escritório da entidade certificadora.

Nem todas as entidades certificadoras podem ser confiadas porque é possível que pessoas de má fé se apresentem como pertencentes a uma entidade certificadora que não existe ou que é falsa. Para se confiar numa entidade certificadora,

esta tem que ser mundialmente reconhecida e aprovada.

No mundo da segurança digital as entidades certificadoras aprovadas mundialmente dependem de políticas muito estritas e procedimentos para emitir certificados e, graças a isso, elas mantêm a confiança dos seus clientes. Para uma maior segurança, estas entidades usam obrigatoriamente hardware especial que garante a impossibilidade de fugas de informação importante, como por exemplo, as chaves privadas.

Entre as mais conhecidas entidades certificadoras mundiais encontram-se as seguintes companhias:

VeriSign Inc., Thawte Consulting, GlobalSign NV/SA, Baltimore Technologies, TC TrustCenter AG, Entrust Inc., etc.

Todas as entidades certificadoras possuem certificados e as correspondentes chaves privadas com as quais assinam os certificados emitidos aos seus clientes. Uma entidade certificadora pode ser raiz (*root* CA) ou num nível subsequente. As entidades certificadoras raiz emitem certificados para elas próprias no início da sua actividade, e assinam-no com o mesmo certificado. Estes certificados são chamados Certificados Raiz.

Os certificados Raiz de entidades certificadoras de confiança mundiais estão disponíveis publicamente nos seus sites Web e podem ser usados para a verificação de outros certificados. As entidades certificadoras intermédias dependem de alguma autoridade de um nível superior para lhes emitir um certificado, que lhes permite emitir e assinar certificados para os seus clientes.

É tecnicamente possível usar qualquer um certificado para assinar qualquer outro, mas na prática a possibilidade de assinar certificados está altamente limitada. Todos os certificados contêm informação que não pode ser alterada sobre se podem ser usados para assinar outros certificados. As entidades certificadoras emitem certificados para os seus clientes, e estes certificados não podem ser usados para assinar outros certificados.

Os certificados que podem ser usados para assinar outros certificados são emitidos só a entidades certificadoras com precauções de segurança muito fortes. Se um cliente obtém um certificado de uma entidade certificadora e assina outro certificado com ele, o novo certificado assinado será inválido porque é assinado por um certificado no qual está especificado que não pode ser usado para assinar outros certificados.

Um determinado certificado pode ser assinado por outro certificado (mais frequentemente, a propriedade de alguma entidade certificadora) ou ser assinado por

si próprio. Os certificados que não são assinados por outro certificado, mas sim por si próprios, são chamados certificados auto-assinados. Especialmente, os Certificados Raiz das entidades certificadoras raiz são certificados auto-assinados. Geralmente, um certificado auto-assinado não pode certificar a relação entre uma chave pública e uma determinada pessoa porque, usando o software apropriado, qualquer um pode criar tal certificado para o nome da pessoa escolhida ou companhia.

Apesar dos certificados auto-assinados não poderem ser confiáveis, eles têm a sua própria aplicação. Por exemplo, dentro das fronteiras da infra-estrutura de uma companhia, onde é possível transferir certificados fisicamente de um modo seguro entre empregados individuais e os sistemas da companhia, certificados auto-assinados são um bom substituto de certificados emitidos por entidades certificadoras.

Numa companhia deste género, não é necessário que uma entidade certificadora confirme que uma determinada chave pública pertence a uma pessoa em particular, porque isto pode ser garantido pelo método de emissão e transferência de certificados. Por exemplo, quando uma nova pessoa é empregada por uma determinada companhia, é possível para o administrador do sistema emitir, para si próprio, um certificado auto-assinado e fornecer esse certificado ao novo empregado de forma segura.

Assim o administrador pode transportar o certificado para todos os sistemas da companhia, de maneira segura e desta forma garante que todos os sistemas internos têm os certificados reais de todos os empregados.

O esquema de segurança baseado em certificados auto-assinados pode ser melhorado se a companhia estabelecesse a sua própria autoridade de certificação local para os seus empregados. Para essa finalidade, a companhia deve inicialmente emitir um certificado auto-assinado para depois emitir certificados assinados com esse certificado para os seus empregados. Dessa forma, o certificado inicial da companhia é um certificado Raiz de confiança e a companhia é, ela própria, uma entidade certificadora raiz.

Em ambos os esquemas descritos, existe a possibilidade de mau uso pelo administrador do sistema que tem os direitos para emitir certificados. Este problema pode ser resolvido reforçando os procedimentos estritos da companhia para a emissão e controlo dos certificados, mas a segurança completa não pode ser garantida.

Em comunicações na Internet, onde não há maneira segura de determinar se um dado certificado enviado pela rede foi ou não alterado algures pelo caminho,

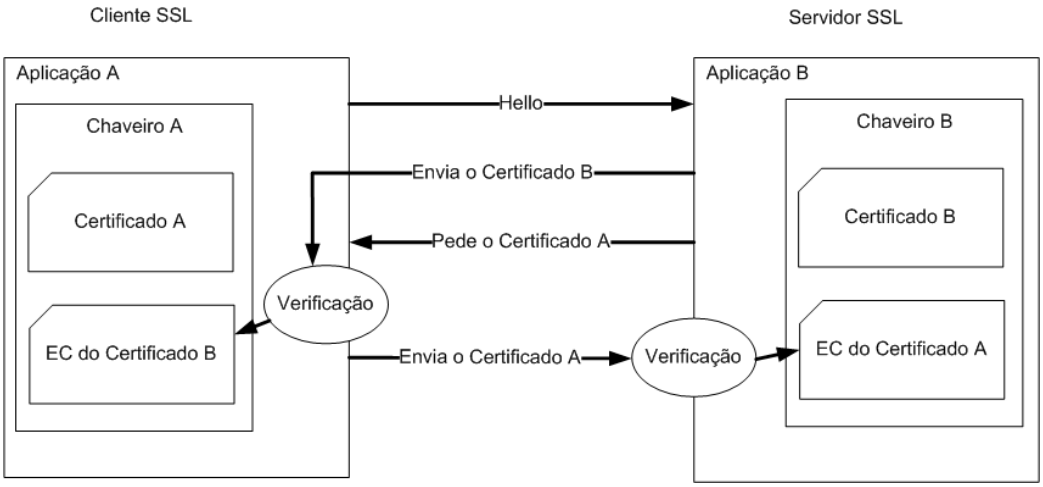

Figura 2.2: SSL - Autenticação Mútua

os certificados auto-assinados praticamente não são usados, apenas certificados emitidos por alguma entidade certificadora aprovada. Em tais redes, o protocolo SSL é mais frequentemente usado para assegurar as comunicações fornecendo canais seguros, chamados túneis SSL.

O protocolo SSL (Secure Socket Layer) baseia-se em criptografia de chave pública e certificados para permitir a comunicação entre duas entidades e que elas criem um canal cifrado de comunicação entre si. Ele garante que o canal é seguro apenas se os certificados usados na comunicação forem de confiança. Por exemplo, se um servidor Web na internet tiver que comunicar com *browsers* Web sobre um canal de comunicação seguro (túnel SSL), deve possuir um certificado emitido por uma entidade certificadora bem conhecida. De outra forma, seria possível que um canal de comunicações entre os clientes e o servidor Web fosse escutado por pessoas de má fé.

A Figura 2.2 descreve o modelo de autenticação mútua entre um cliente e um servidor. A aplicação do cliente verifica a identidade do servidor e a aplicação do servidor verifica a identidade do cliente.

Os certificados emitidos por entidades certificadoras aprovadas permitem segurança de alto nível na comunicação, independentemente de serem usados numa rede corporativa privada ou na internet. No entanto, os certificados auto-assinados por vezes são usados porque os certificados emitidos por entidades certificadoras custam dinheiro e requerem esforços em nome do seu dono para a emissão inicial, renovação periódica e fiabilidade no armazenamento da chave privada correspondente.

2.2.5 Cadeias de Certificação e Verificação

Quando uma entidade certificadora topo-de-nível emite um certificado para um cliente, ela assina-o com o seu certificado Raiz. Desta maneira, é criada uma cadeia de certificação formada por dois certificados, o certificado da entidade certificadora que precede o certificado do cliente. Uma cadeia de certificados é a sequência de certificados na qual cada certificado está assinado pelo seu predecessor.

No inicio da cadeia normalmente fica um certificado emitido para um cliente final, e no final da cadeia está o certificado Raiz de uma entidade certificadora. Pelo meio da cadeia estão os certificados de entidades certificadoras intermédias. A prática comum é que as entidades certificadoras topo-de-nível emitam certificados para as entidades certificadoras intermédias e que especifiquem nestes certificados que eles podem ser usados para emitir outros certificados.

As entidades certificadoras intermédias emitem certificados para os seus clientes ou para outras entidades certificadoras intermédias. Os direitos atribuídos ao cliente final não os permitem utilizar os seus certificados para assinar outros certificados, mas esta restrição não se aplica a entidades certificadoras intermédias.

Um certificado no inicio da cadeia de certificação pode ser de confiança só se a sua cadeia de certificação for verificada com sucesso. Neste caso, diz-se que se trata de um certificado verificado. A verificação de uma cadeia de certificação verifica se todos os certificados na sua cadeia são assinados pelo certificado seguinte na cadeia. Quanto ao último certificado, é verificado se ele se encontra numa lista de certificados Raiz de confiança.

Qualquer sistema de software que efectue verificação de certificados mantém uma lista de certificados Raiz de confiança, na qual confia incondicionalmente. Estes são os certificados Raiz das entidades certificadoras mundialmente reconhecidas. Por exemplo, o browser Web Internet Explorer, à partida, vem com uma lista de cerca de 150 certificados Raiz de confiança, e o browser Firefox na sua instalação inicial contém cerca de 70 certificados de confiança.

A verificação da cadeia de certificação não inclui apenas a verificação que cada certificado é assinado pelo seguinte e que o certificado no final da cadeia está na lista de certificados Raiz de confiança. Também é necessário verificar que cada certificado na cadeia ainda é válido, e que todos os certificados com excepção do primeiro têm o direito de ser usados para assinar outros certificados. Se a verificação da última condição fosse omissa, seria possível para os clientes finais a emissão de certificados a quem eles quisessem e a verificação do certificado emitido seria sucedida.

Na verificação de uma determinada cadeia de certificados, é verificado também se algum certificado nela foi revogado. A finalidade da combinação de todas as verificações descritas é verificar se um certificado é de confiança. Se a verificação de uma cadeia de certificação falha, não significa que se trata de uma tentativa de falsificação. É possível que a o certificado Raiz da cadeia não se encontre na lista de certificados de confiança.

Geralmente, um certificado não pode ser verificado se a sua cadeia de certificação não está presente ou se o certificado Raiz não se encontrar na lista de certificados de confiança. A cadeia de certificação de um determinado certificado pode ser construída programadamente em caso de não estar presente, mas para isso, todos os certificados nela têm que estar presentes.

2.2.6 Chaveiros Protegidos e Ficheiros Certificados

Em sistemas com a assinatura electrónica de documentos, é usado armazenamento protegido para chaves e certificados (chaveiro protegido). Este armazenamento pode conter três elementos, que são: certificados, cadeias de certificação e chaves privadas. Como a informação guardada em armazenamento protegido é confidencial devido a considerações de segurança, ela é acedida usando palavras-chave de dois níveis, uma palavra-chave para o chaveiro e palavras-chave diferentes para cada chave nele guardadas. Graças a estas palavras-chave, em caso de um eventual roubo de um chaveiro protegido, a informação que ele contém não pode ser facilmente lida.

Na prática, as chaves privadas, como uma particularidade importante e peça confidencial de informação, nunca são armazenadas fora de chaveiros e estão sempre protegidas com uma palavra-chave de acesso.

Existem várias normas desenvolvidas para chaveiros protegidos. A mais divulgada é a norma PKCS#12, na qual o armazenamento é um ficheiro com a extensão .PFX (ou a extensão usada mais raramente .P12). Um ficheiro PFX normalmente contém um certificado, uma chave privada correspondente a ele, e uma cadeia de certificação provando a autenticidade do certificado.

A presença de uma cadeia de certificação não é necessária e muitas vezes os ficheiros PFX apenas contêm o certificado e a chave privada. Na maior parte dos casos, para facilitar o utilizador, a palavra-chave para aceder a um ficheiro PFX é igual à palavra-chave de acesso à chave privada nele armazenada. Por esta razão, quando se usam ficheiros PFX muito frequentemente, apenas uma palavra-chave de acesso é necessária.

Quando uma entidade certificadora emite um certificado digital a um cliente, o cliente recebe um chaveiro protegido, que contem o certificado emitido, a sua correspondente chave privada e toda a cadeia de certificação, provando a autenticidade do certificado. O armazenamento protegido é fornecido ao cliente, ou em forma de ficheiro PFX ou de *smart card*, ou é directamente instalado no seu *Web browser*.

Normalmente, quando um certificado é emitido pela Internet, independentemente de como o utilizador confirma a sua identidade, no procedimento de emissão o *browser Web* do utilizador tem um papel importante. Num pedido de emissão de certificado enviado a uma certa entidade certificadora do seu Web site, o *browser Web* do utilizador gera um par de chaves pública/privada e envia a chave pública para o servidor da entidade certificadora. O *browser* guarda secretamente a chave privada e não a envia para a entidade certificadora. A entidade certificadora, após verificar a autenticidade dos dados pessoais do seu cliente, emite-lhe um certificado no qual grava a chave pública recebida pelo *browser Web* do cliente e os seus dados de identidade confirmados.

Para alguns tipos de certificados, os dados de identidade podem consistir apenas num endereço de e-mail verificado, enquanto para outros os dados podem conter informação completa sobre a pessoa: nome, endereço, número de bilhete de identidade, etc. A verificação dos dados pessoais é feita usando um procedimento, determinado pela respectiva entidade certificadora. Depois do servidor da entidade certificadora emitir o certificado ao seu cliente, ele redireciona-o para a página Web da qual este certificado pode ser instalado no *browser Web* do cliente.

Na realidade, o utilizador recebe de alguma forma o seu novo certificado da entidade certificadora junto com a cadeia de certificação completa. Entretanto, o *browser Web* armazenou a chave privada correspondente ao certificado e, por fim, o utilizador obtém um certificado e a sua correspondente chave privada, junto da cadeia de certificação do certificado, instalado no seu *Web browser*.

O método de armazenamento de chaves privadas varia com os diferentes *browsers*, mas em qualquer dos casos essa informação confidencial é protegida pelo menos com uma palavra-chave. No mecanismo descrito para emissão de um certificado, a chave privada do utilizador permanece desconhecida para a entidade certificadora e dessa maneira o utilizador pode ter a certeza de que mais ninguém tem acesso à sua chave privada.

A maior parte dos *browsers Web* conseguem usar os certificados e chaves privadas armazenadas neles para autenticação antes de assegurar os servidores SSL. Muitos clientes de e-mail também usam os certificados armazenados nos *browsers Web* para assinar, cifrar, e decifrar correio electrónico. No entanto, al-

gumas aplicações não conseguem usar directamente os certificados a partir dos *browsers Web* dos utilizadores, mas conseguem lidar com chaveiros PFX. Nestes casos, os utilizadores podem exportar os seus certificados a partir dos *browsers Web*, juntamente com as respectivas chaves privadas para ficheiros PFX, e assim usá-los noutra aplicação.

No Internet Explorer, o certificado e a chave privada são exportados a partir do menu principal usando os comandos Ferramentas — Opções da Internet — Conteúdo — Certificados — Exportar; e no Firefox usando os comandos Ferramentas — Opções — Avançado — Cifra — Ver certificados — Exportar. À partida, quando se exporta um certificado e a sua chave privada com o Internet Explorer para um ficheiro PFX, a cadeia de certificação não é completamente incluída no ficheiro de saída, mas o utilizador pode especificar que a quer numa opção adicional.

Há várias normas de armazenamento de certificados digitais X.509. A mais comum é a codificação ASN.1 DER, na qual os certificados são armazenados em ficheiros com extensão .CER (em raras situações com extensão .CRT, .CA, ou .PEM). Um ficheiro CER contém uma chave pública, informação sobre o seu dono e uma assinatura digital de alguma entidade certificadora, certificando que esta chave pública pertence mesmo a esta pessoa. Um certificado no formato .PEM não é mais do que um certificado no formato DER, codificado em Base64, dentro de duas linhas ––BEGIN CERTIFICATE–– e ––END CERTIFICATE––. O tamanho ocupado por um certificado .PEM é cerca de 4/3 do tamanho de um certificado DER.

As entidades certificadoras divulgam a partir dos seus sites os seus certificados Raiz em ficheiros CER. Um ficheiro CER pode ser armazenado em formato binário ou em formato de texto, codificado com Base64.

2.2.7 Revogação de Certificados

Às vezes, acontece a uma pessoa ou a uma companhia perder o controlo sobre os seus certificados e as chaves privadas correspondentes, e estes caem nas mãos de outras pessoas, que podem eventualmente tirar proveito deles. Nestes casos é necessário revogar estes certificados (certificados revogados).

As entidades certificadoras periodicamente (ou em caso de emergência) publicam listas de certificados que estão temporariamente inactivos ou revogados antes da sua data de expiração. Estas listas são assinadas digitalmente pela entidade certificadora que as emite, e são chamadas de listas de certificados revogados

(*Certificate Revogation Lists* - CRL).

Nestas listas são especificados o nome da entidade certificadora que emitiu o certificado, a data de emissão, a data da próxima publicação de uma nova lista, os números de série dos certificados revogados e as vezes específicas e razões para a revogação.

2.2.8 Autenticação Mútua (Cliente e Servidor)

A autenticação mútua entre duas entidades é a capacidade de cada uma das entidades se poder certificar que a outra é quem diz ser. Normalmente é usada quando um cliente se quer autenticar perante um servidor e este também se autentica perante o cliente de maneira a ambas terem a certeza da identidade do outro. Quando se lida com certificados de cliente contidos num *smart card* normalmente é apresentada uma lista de certificados que podem ser utilizados, em que o cliente escolhe o mais adequado, e introduz o código PIN do *smart card*.

2.3 PKCS#11

A PKCS#11 [4] é uma norma que especifica uma API (*Application Programming Interface* - Interface de Programação de Aplicações) criada pelos RSA Laboratories e denominada por Cryptoki para dispositivos criptográficos que executam funções criptográficas. A versão actual é a PKCS#11 v2.20 mas existe um *draft* para a v2.30.

O software distribuido juntamente com os dispositivos criptográficos como por exemplo *smart cards*, normalmente inclui uma implementação da norma PKCS#11 para o cartão e leitor específico. Essa implementação costuma ser uma biblioteca (.dll em Windows e .so em Linux e UNIX) que pode ser carregada dinamicamente e usada por qualquer aplicação instalada na máquina local.

Nem todas as funções definidas na norma PKCS#11 são implementadas pelos vendedores de dispositivos criptográficos, a maior parte opta por implementar apenas as funções necessárias que permitem o uso adequado dos dispositivos. Por exemplo, se um dispositivo é inicialmente concebido com certificados e chaves privadas respectivas em que não é suposto criar novas chaves ou certificados, normalmente são omitidas as funções que permitem a criação de chaves ou certificados novos.

A norma PKCS#11 não permite a extracção física das chaves privadas contidas

no *smart card*, mas é possível usar essas chaves privadas para cifrar, decifrar, ou assinar dados. Mas para que uma operação que use a chave privada seja executada, o utilizador necessita de introduzir o seu código PIN à partida. É isto que protege o acesso ao cartão.

A norma PKCS#11 fornece uma interface para aceder a chaves protegidas e chaveiros que contêm certificados localizados num *smart card*. Por isso o seu modo de operação é semelhante ao modo de operação da PKCS#12 com chaveiros. No entanto os *smart cards* possuem algumas propriedades embutidas que um ficheiro PKCS#12 não tem, como por exemplo a capacidade de efectuar cifra ou assinatura.

2.4 Cartão de Cidadão

O Cartão de Cidadão é o documento que inclui o número de identificação civil, o número de identificação fiscal, o número de utente dos serviços de saúde e o número de identificação da segurança social. Além disso, contem também dados relevantes a cada cidadão para a sua identificação. Alguns destes dados são os mesmos contidos no Bilhete de Identidade com excepção da data de emissão e do estado civil. Também serve como cartão de viagem dentro da União Europeia e outros países no âmbito de convenções internacionais [7].

2.4.1 Como é Constituido o Cartão de Cidadão

Este cartão possui um chip que contem informação específica sobre o seu titular. Com excepção da assinatura digitalizada, todas as informações contidas no exterior do cartão também são contidas no chip. As Figuras 2.3 e 2.4 descrevem cada campo contido no Cartão de Cidadão [8].

O chip ainda inclui as seguintes funcionalidades:

- **IAS** - aplicação responsável pelas operações de autenticação e assinatura electrónica
- **EMV-CAP** - aplicação responsável pela gestão de palavras-chave únicas por canais alternativos
- **Match-On-Card** - aplicação responsável pena informação biométrica de impressões digitais

Figura 2.3: Cartão de Cidadão - Frente

Figura 2.4: Cartão de Cidadão - Verso

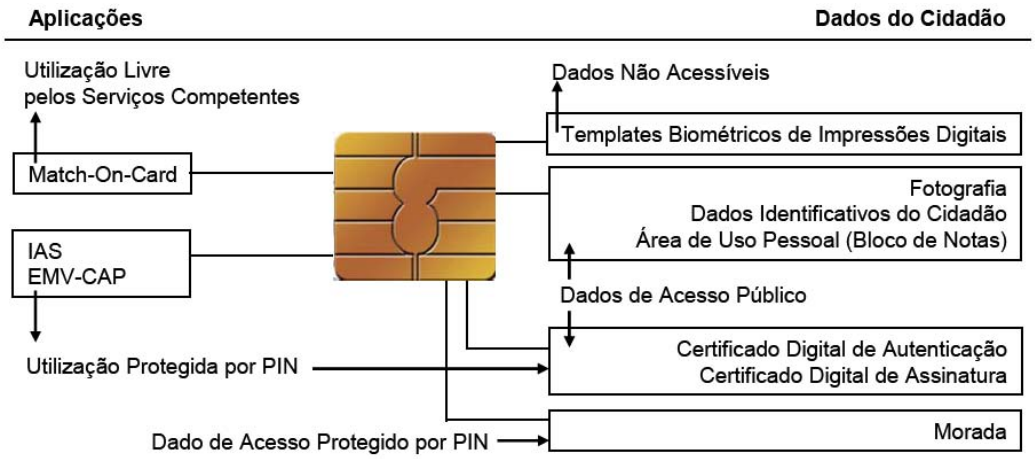

Figura 2.5: Informação e Aplicações Residentes no Chip

A Figura 2.5 descreve a informação e as aplicações residentes no chip do Cartão de Cidadão.

O Cartão possui um espaço onde podem ser guardados dados pessoais, designado por "Área Pessoal" ou por "Bloco de Notas". Esse espaço pode conter cerca de 1KB (1000 caracteres).

2.4.2 Autenticação Simples com o Cartão de Cidadão

Para que o servidor possa efectuar o pedido do certificado existente no CC, é preciso efectuar os seguintes passos:

- Configurar o servidor para ligações SSL → o site em questão deve possuir um certificado instalado no servidor Web para que seja possível estabelecer uma ligação segura entre o servidor e as aplicações clientes.
- Configurar o servidor para aceitar certificados de clientes → configura-se o servidor para que efectue o pedido de um certificado digital ao cliente.
- Configurar o servidor para pedir e aceitar o certificado do Cartão de Cidadão → Normalmente os servidores estão configurados para pedir e aceitar certificados clientes emitidos pelo seu LDAP (*Lightweight Directory Access Protocol*). Aqui configura-se o servidor para que também aceite certificados emitidos pelas Entidades Certificadoras emissoras dos certificados presentes no Cartão de Cidadão.

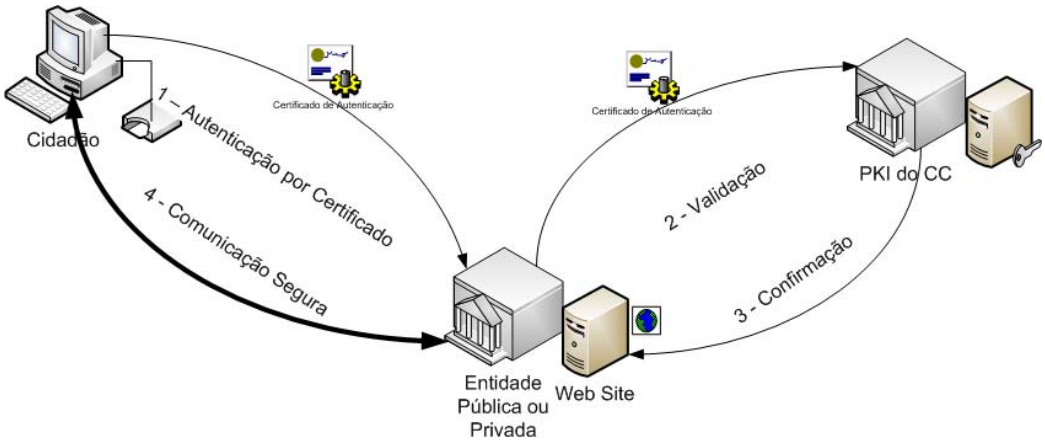

Figura 2.6: Autenticação Genérica com o Cartão de Cidadão

- Validação aplicacional do certificado → Para garantir que só são aceites certificados presentes no CC, o código desenvolvido para autenticação deve validar um conjunto de parâmetros presentes no certificado, para garantir a sua origem.

- Validação de validade do certificado → A última validação é feita pela entidade emissora do certificado, para garantir que o certificado não foi revogado, podendo ser usados dois métodos para esse efeito, CRL (*Certificate Revocation List*) ou OCSP (*Online Certificate Status Protocol*).

O processo de validação garante a associação entre os dados do CC e o seu titular. A verificação do certificado digital na Entidade Certificadora permite verificar se o certificado se encontra válido. No entanto, cabe à Entidade que, após a verificação no seu sistema de informação, a definição dos privilégios de acesso para o utilizador.

Este modelo de autenticação não é o suficiente para autenticar um profissional de saúde na RTS, porque o certificado de autenticação do Cartão de Cidadão não tem informação relativa ao cargo desempenhado por um determinado profissional numa Instituição de Saúde. A criação de um novo certificado com essa informação é uma necessidade, e torna-se obrigatória uma redefinição da arquitectura.

A Figura 2.6 descreve como funciona o mecanismo de autenticação usando o certificado de autenticação do Cartão de Cidadão.

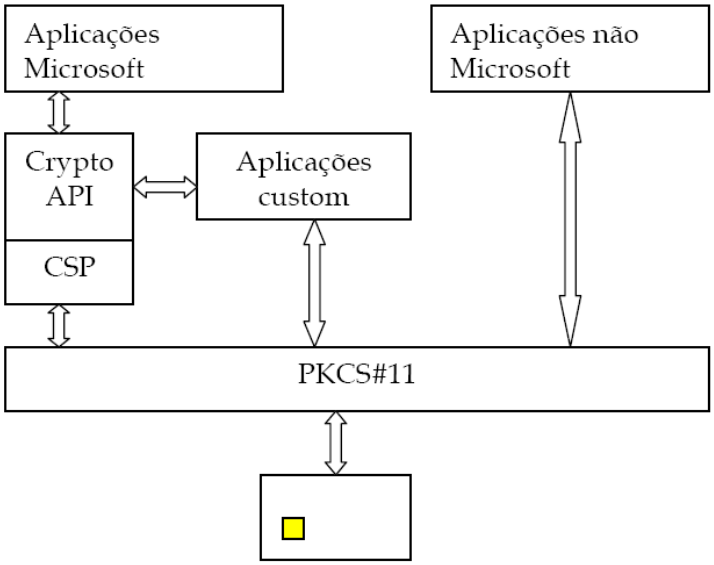

Figura 2.7: Interacção entre aplicações e o módulo PKCS#11

2.4.3 Middleware do Cartão de Cidadão

O Cartão de Cidadão é fornecido com uma interface PKCS#11 que é utilizada por aplicações não Microsoft. Para aplicações Microsoft seria usada a interface CSP. A interface CSP não foi analisada porque apenas está disponível em ambientes Windows.

A Figura 2.7 descreve as diferenças na interacção do *browser* Internet Explorer (Aplicação Microsoft) e o *browser* Firefox (Aplicação não Microsoft).

A interface PKCS#11 implementa as seguintes funções:

Funções Gerais:

- C_Initialize
- C_Finalize
- C_GetInfo
- C_GetFunctionList

Funções de gestão de Slot e Token:

- C_GetSlotList
- C_GetSlotInfo
- C_GetTokenInfo
- C_GetMechanismList
- C_GetMechanismInfo
- C_WaitForSlotEvent (only non-blocking)
- C_SetPin

Funções de gestão de sessão:

- C_OpenSession
- C_CloseSession
- C_CloseAllSessions
- C_GetSessionInfo
- C_Login
- C_Logout

Funções de gestão de objectos:

- C_FindObjectsInit
- C_FindObjects
- C_FindObjectsFinal
- C_GetAttributeValue

Funções de assinatura:

- C_SignInit
- C_Sign

- C_SignUpdate
- C_SignFinal

Funções de digest:

- C_DigestInit
- C_Digest
- C_DigestUpdate
- C_DigestFinal

Funções de geração aleatória:

- C_SeedRandom
- C_GenerateRandom

As restantes funções definidas na norma PKCS#11 que não aparecem na lista não foram implementadas pela interface fornecida.

Os mecanismos de assinatura suportados pelo cartão são:

Para assinaturas:

- CKM_RSA_PKCS: ambos ASN.1-wrapped e hashes puros (MD5, SHA1, SHA1+MD5, RIPEMD160)

- CKM_RIPEMD160_RSA_PKCS, CKM_SHA1_RSA_PKCS, CKM_MD5_RSA_PKCS

Para digests:

- CKM_SHA_1, CKM_RIPEMD160, CKM_MD5

Os objectos incluidos no Cartão de Cidadão que podem ser manipulados pelas funções implementadas pelo módulo da PKCS#11 fornecido com o *middleware* são descritos na Tabela 2.4.3, em que a primeira coluna refere o *handle* do objecto e a segunda a sua designação.

1	Authentication Private Key
2	Authentication Certificate
3	Authentication Public Key
4	Authentication Sub CA Certificate
5	Authentication Sub CA Public Key
6	Signature Private Key
7	Signature Certificate
8	Signature Public Key
9	Signature Sub CA Certificate
10	Signature Sub CA Public Key

Tabela 2.1: Objectos Existentes no Cartão de Cidadão

2.4.4 API eID Lib

Esta aplicação é dedicada a organizações cujo objectivo é desenvolver aplicações que utilizam o Cartão de Cidadão. Este kit de desenvolvimento lida apenas com dados identificativos do cartão, incluindo o Bloco de Notas, e não com operações criptográficas.

2.4.5 Normas dos Leitores de Cartões

O Cartão de Cidadão segue as normas internacionalmente recomendadas encontrando-se alinhado pelas orientações correntes da União Europeia, nomeadamente as do grupo de trabalho para o *European Citizen Card* (ECC), pelas normas definidas pela *International Civil Aviation Organization* (ICAO) para documentos de viagem internacionais (documento 9303) e os as normas 7501 e 7810 definidas pela *International Organization for Standardization* (ISO).

Os leitores devem estar de acordo com as seguintes normas [9]:

- PC/SC versão 1.0;
- ISO/IEC 7816-1,2,3,4: IC Cards with Contacts;
- EMV Level 1;
- CT-API versão 1.1;
- CCID - Chip Card Interface Device 1.0;

- Suportar a norma ISO/IEC 7186 Class A, B e C (*smart cards* com voltagens de 5V, 3V, 1.8V);

- Suportar leitura e escrita para *smart cards* com microprocessadores alinhados com ISO/IEC 7816-1,2,3,4, protocolos T=0 e T=1;

- Suportar *smart cards* com frequências de relógio até 8Mhz;

- **Microsoft Windows Hardware Quality Labs (WHQL)** - caso o sistema operativo seja Windows;

- **Microsoft Windows Logo Program WLP 2.0** - caso o sistema operativo seja Windows;

- **EMV Level 1 (EMV2000)**;

Os leitores deverão suportar *smart cards* de formato ID-1.

2.4.6 Cartão de Cidadão e *Smart Card* Genérico

O Cartão de Cidadão vem com 2 certificados de origem, e respectivos pares de chaves publica/privada. Não é possível definir novos objectos no Cartão de Cidadão. Num *smart card* é possível criar novos objectos, certificados e pares de chave tendo apenas como limite o espaço físico disponível.

A criação de novos objectos no Cartão de Cidadão é impedida devido à função *C_Create_Object* não ter sido implementada. Obviamente que a falta de espaço disponível também é uma limitação, mas mesmo sem espaço disponível, a existência da função *C_Create_Object* iria permitir a criação de objectos ao nível da sessão, caso tivesse sido implementada.

Também não é possível a alteração das credenciais já existentes no cartão porque essas credenciais se tratam de certificados assinados, e qualquer alteração seria detectada durante o processo de verificação dos próprios, pois eles estão digitalmente assinados pela Entidade Certificadora que os emitiu. Adicionar extensões a certificados existentes está fora de questão.

2.5 O Projecto OpenSSL

O Projecto OpenSSL[1] é um esforço colaborativo para desenvolver um conjunto de ferramentas completo a nível comercial e em código aberto dos protocolos SSL (*Secure Sockets Layer*) e TLS (*Transport Layer Security*) como também uma biblioteca criptográfica.

O projecto é gerido por uma comunidade a nível mundial de voluntários que usam a Internet para comunicar, planear, e desenvolver ferramentas para o OpenSSL.

O OpenSSL possui ferramentas que permitem criar pedidos de certificados (CSR - *Certificate Signing Request*) de uma forma fácil apenas executando um comando numa consola. Também é possível a criação de Entidades Certificadoras usando o conjunto de ferramentas fornecido pelo OpenSSL. Essas entidades podem validar pedidos de certificados, emitindo-os em vários formatos diferentes, como por exemplo DER ou PEM.

Como se trata software de código aberto, tem a vantagem de se poder adaptar a outras situações sendo sujeito a alterações no seu código.

[1] http://www.openssl.org/

Capítulo 3

Trabalhos Relacionados

Este capítulo descreve alguns projectos semelhantes ao projecto da RTS em que também são usados *smart cards* em infra-estruturas de chaves públicas para a autenticação em redes telemáticas de saúde em vários países. Também é feita uma breve descrição da arquitectura actual da RTS com o uso de *smart cards* e de vários aspectos que devem ser considerados para a sua adaptação ao uso do Cartão de Cidadão.

3.1 Alemanha - Electronic Health Card (eHC) e Health Professional Card (HPC)

Durante os próximos anos o cartão de saúde alemão vai ser substituído por um novo cartão de saúde (eHC). A introdução deste novo cartão visa melhorar a eficiência do sistema de saúde e dos direitos dos pacientes. Para reduzir os custos do sector público e criar bases de comunicação homogéneas foi criado um sistema a nível nacional, a Infra-estrutura Telemática de Saúde.

O eHC não só contém dados administrativos mas também informação detalhada sobre o paciente e os seus tratamentos. Esta informação é pessoal e está sob a obrigação de segredo imposta pela relação médico/paciente e protegida por lei, é agora armazenada numa base de dados central em função de melhorar os serviços para os pacientes[10].

Os profissionais de saúde possuem um cartão diferente, denominado por *Health Professional Card* (HPC), com funcionalidades especiais para os profissionais de saúde. O processo de criação é semelhante ao do eHC, em que é gerado um certifi-

Figura 3.1: Cartão de Saúde do Profissional (HPC)

cado e a sua chave privada fica armazenada num ambiente seguro. Possui funções CVC (*Card-Verifiable Certificate*) que validam os dados pessoais do profissional contidos no cartão, para que estes não possam ser forjados. A Figura 3.1 apresenta o HPC.

Tal como a RTS, esta rede telemática de saúde também se trata de uma rede a nível nacional. Foi criado um projecto que visa introduzir o eHC na Alemanha. A infra-estrutura telemática de saúde é composta por uma parte central, que consiste em centros de dados com bases de dados centralizadas, e as partes periféricas, que consistem em utilizadores do sistema, como por exemplo o profissional de saúde, farmácias ou hospitais. Ambas as partes estão ligadas por um túnel VPN.

O eHC, mostrado na Figura 3.2 tem o mesmo tamanho de um cartão de crédito comum, à frente tem informação relativa ao indivíduo a que pertence, e um *micro-chip*. O eHC é um *smart card*, o que significa que contém um processador próprio, com o seu próprio conjunto de instruções. É isto que o distingue do cartão actual, que apenas é um cartão de memória.

É possível emitir receitas electrónicas com este cartão, carregando a receita para o chip e assinando-a electronicamente. Essa informação pode ser lida na farmácia, por exemplo.

Este projecto envolve uma das mais abrangentes infra-estruturas de chaves públicas do mundo. A companhia encarregada de desenvolver o projecto é a *Gematik*[1].

[1] http://www.gematik.de/

Figura 3.2: Cartão de Saúde do Utente (eHC)

3.2 Uso de *Smart Cards* em Redes Telemáticas de Saúde

Tal como a Alemanha, muitos outros países optaram pelo uso de *smart cards* nas suas redes telemáticas de saúde. Alguns dos mais populares, neste momento, além da Alemanha são a Algeria, Áustria, Austrália, Bélgica, França, Hungria, Itália, México, Eslovénia, Espanha, Tailândia e Reino Unido, sendo a sua maioria produzidos pela Gemalto[2].

No geral, o acesso é baseado usando uma infra-estrutura de chaves públicas, da mesma maneira que é descrita na secção seguinte, que apresenta uma arquitectura para a RTS com o uso de *smart cards*.

3.3 Arquitectura Definida pelo H. Gomes

Esta arquitectura é a que serve de base para a adaptação ao uso do Cartão de Cidadão na RTS, pois foi pensada para ser usada com *smart cards*. Segue-se uma breve descrição, que entra em detalhes mais técnicos no Capítulo 6, onde são descritos os aspectos que devem ser alterados para o seu funcionamento com o Cartão de Cidadão.

[2] www.gemalto.com

3.3.1 Introdução

Esta arquitectura baseia-se numa infra-estrutura de chaves públicas, de modelo híbrido, onde cada Instituição de Saúde (IS) é responsável pela emissão e gestão de certificados digitais para as suas entidades. Para isso, a RTS e cada uma das IS, constitui a sua própria PKI. Entre as PKI das várias IS e a PKI da RTS são estabelecidas relações de confiança, baseadas em certificação cruzada.

Para o transporte e armazenamento seguro das credenciais de autenticação dos Profissionais são utilizados *smart cards*. Estes cartões permitem um nível de autenticação forte, baseado em dois factores, posse e senha, e além disso um elevado nível de protecção física dos dados guardados do seu interior. Também podem ser facilmente transportados pelo seu dono.

As credenciais de autenticação dos Profissionais são constituidas por dois certificados de chave pública e correspondentes chaves privadas: (i) um certificado para o acesso à RTS e (ii) um certificado para a autenticação do Profissional na renovação do seu certificado RTS perante a IS. O certificado RTS cumpre as funções de acesso à RTS e indicação do seu perfil na IS a que está vinculado.

Devido às alterações de perfil poderem ser alteradas frequentemente, é definido um tempo de vida curto para os certificados RTS. Este tempo de vida curto permite a implementação de uma PKI simplificada, pois deixa de ser obrigatória a existência de CRLs na autenticação para o acesso à RTS.

No modelo de confiança utilizado, cada entidade tem como âncora de confiança exclusivamente a EC raiz da instituição a que está vinculada. Além disso como os caminhos de certificação são curtos, o número de certificados de confiança a que cada entidade necessita de ter na sua posse, para validar os certificados que lhe são apresentados, é reduzido, o que é uma vantagem dado o pouco espaço de memória nos *smart cards*.

3.3.2 Descrição da Arquitectura

Esta secção descreve a arquitectura proposta pelo H. Gomes [5] A solução apresentada por esta arquitectura baseia-se em pares de chaves assimétricas e certificados digitais das chaves públicas. Esta tecnologia é suportada pelos navegadores servidores Web mais populares.

É utilizada uma infra-estrutura de chave pública, PKI (*Public Key Infrastructure*). É proposta uma PKI simplificada baseada numa arquitectura previamente proposta para resolver o problema de autenticação de utilizadores em conjuntos

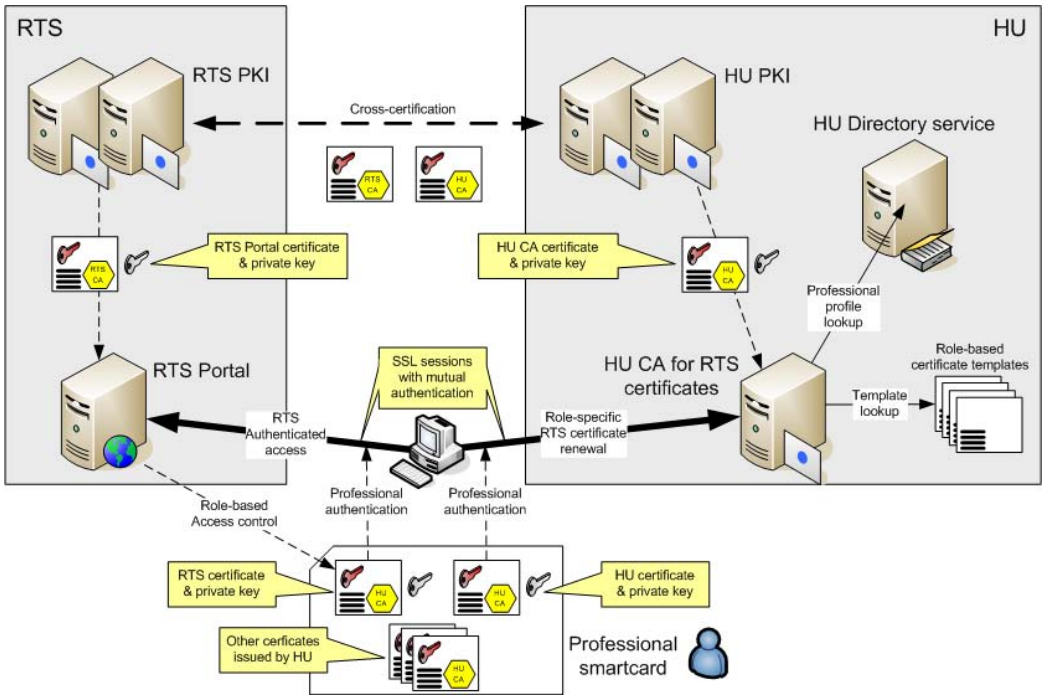

Figura 3.3: Visão Geral da Arquitectura da RTS

de instituições universitárias. Esta arquitectura baseia-se em certificados digitais com períodos de validade de curta duração e certificação cruzada entre instituições para a validação de cadeias de certificação. A Figura 3.3 apresenta uma visão geral da arquitectura da RTS [11].

São utilizados *smart cards* para o armazenamento e protecção das chaves privadas e certificados raiz confiáveis. Estes *smart cards* são utilizados apenas na autenticação dos Profissionais, para os Utentes a autenticação é feita através de um nome de acesso e uma senha.

O Uso de Certificados

A RTS é uma aplicação que disponibiliza os seus serviços aos utilizadores através de Portais com interface Web. Esses serviços necessitam da autenticação do Profissional, da instituição a que pertence, e qual o seu cargo nessa instituição. Os mecanismos disponíveis para autenticação do profissional utilizando um navegador (*browser*), em servidores Web são o nome e senha, ou certificados digitais.

O método mais vulgar é o da utilização de nome e senha. Este método tem

várias vulnerabilidades, porque geralmente, ou as senhas são demasiado fracas para facilitar a memorização, ou são demasiado complexas e necessitam de ser anotadas, o que as pode comprometer. Além disso, para o Portal obter a restante informação sobre o utilizador, seria necessário um serviço com a informação de todos os utilizadores da RTS, ou obter a informação de um determinado utilizador na sua instituição de origem.

A primeira situação estaria a replicar informação de directoria de todas instituições participantes na RTS. A segunda implicaria que a RTS tivesse acesso online à informação de directoria das várias Instituições de Saúde.

O uso de certificados permite que a autenticação e sua validação sejam feitas localmente. Para isso, é necessário que o certificado possua a informação do seu proprietário, informação sobre a instituição de origem e do cargo nela desempenhado, e de um período de validade que permita prescindir da verificação da sua revogação.

Se a chave privada e o correspondente certificado digital forem armazenados num *smart card* protegido com um código de acesso, a segurança torna-se mais eficiente, uma vez que para efectuar a autenticação é necessária a posse do *smart card*, e de conhecer o seu código de acesso.

Modelo de Comunicação

A Figura 3.4 descreve o modelo de comunicação e autenticação na RTS. O profissional envia o seu certificado ao Portal da RTS. Depois da validação do certificado, em função da informação obtida nele, são aplicadas as autorizações adequadas ao seu perfil.

A comunicação entre o Portal e outros serviços ou servidores é mediada, no sentido em que o Profissional que fez o pedido de informação não se autentica neste troço. O Portal é o responsável pelo pedido e por garantir que não fornece ao Profissional mais informação do que a que ele pode aceder. No entanto continua a haver autenticação das entidades envolvidas na comunicação para garantir que a informação não é acedida por entidades não credenciadas para o acesso. Para aceder à informação pretendida, pode ser necessário recorrer a vários serviços em cadeia sendo aplicado sempre o mesmo modelo de autenticação entre os servidores.

A identificação subadjacente à autenticação é feita através da troca de certificados entre as entidades. Cada entidade valida o certificado da outra para que se possa continuar com a comunicação. O Portal tem que garantir que o utilizador

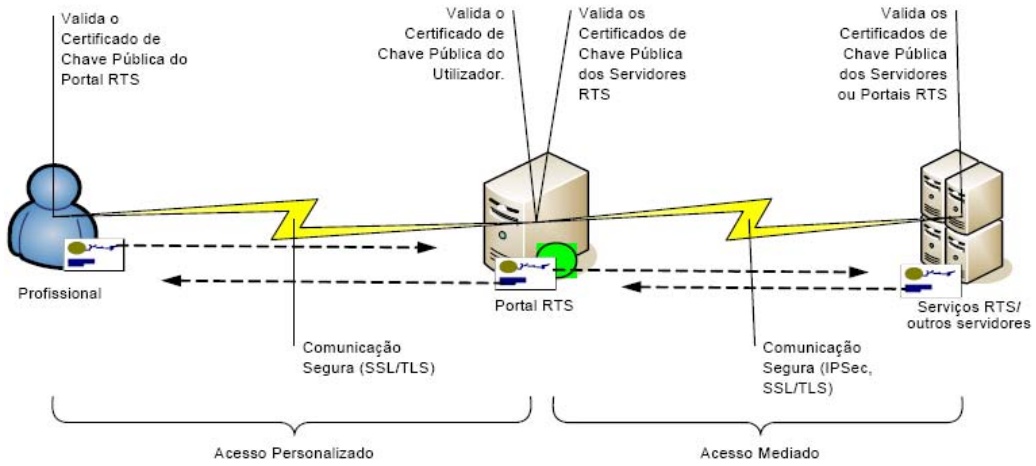

Figura 3.4: Modelo de Comunicação e Autenticação

pertence a um tipo válido. Após a autenticação, é criado um canal de comunicação seguro.

No caso da comunicação entre o Utente e o Portal, em que o Utente não possui um certificado, o modelo é semelhante, mas não há o envio do certificado do Utente para o Portal. No entanto, o Utente recebe um certificado do Portal e deve proceder à sua validação para que possa ser criado um canal de comunicação seguro entre eles. Depois de criado um canal seguro é pedido ao utilizador o nome de acesso e a senha.

A tecnologia utilizada para estabelecer uma ligação entre um Profissional e o Portal RTS é HTTP sobre SSL/TLS (HTTPS), e entre os Servidores institucionais é HTTPS ou IPSec.

Requisitos dos Certificados

Cada entidade, Profissional ou Serviço/Servidor, deve possuir um par de chaves assimétricas e um certificado de chave pública, emitido pela sua instituição de origem, para que se possa autenticar na RTS.

O certificado do Profissional deve conter a identificação do seu possuidor, a sua instituição de origem, e o cargo que desempenha na instituição. É com base nesta informação que são definidas as permissões de acesso à informação gerida pela RTS.

Tipos de Certificados de Profissionais

São utilizados certificados no formato X.509v3. Como para além da identificação da entidade emissora é necessário definir o cargo desempenhado pelo profissional na instituição, é necessário utilizar extensões previstas na versão 3. Desta forma é usada a extensão EKU (*Extended Key Usage*), e definidos OIDs (*Object Identifiers*) para cada tipo de Profissional existente no sistema.

Armazenamento dos Certificados e Chaves Privadas

As técnicas de autenticação são baseadas naquilo que se sabe, naquilo que se possui, naquilo que se é, ou em combinações das três anteriores.

A utilização de *smart cards* para o armazenamento das chaves privadas e respectivos certificados permitem a autenticação baseada em dois factores, a posse do cartão e o conhecimento do código de activação do mesmo. Outra vantagem dos *smart cards* é possuírem um processador criptográfico que permite a geração e utilização do par de chaves no seu interior impedindo que a chave privada saia para o exterior.

Os *smart cards* têm como desvantagem o aumento do custo da implementação do sistema uma vez que é necessário equipar os computadores utilizados pelos Profissionais de leitores de *smart cards*.

Tempo de Vida dos Certificados

Um dos aspectos fundamentais numa PKI é a gestão de listas de certificados revogados (CRL). Para simplificar a gestão da PKI a utilizar na RTS, optou-se pela utilização de certificados com intervalos de tempo curtos para a autenticação dos Profissionais, e assim prescindir de CRLs, ou outros mecanismos, para lidar com a revogação dos certificados dos profissionais. O seu tempo de vida deve resultar de uma ponderação sobre o risco associado ao seu tempo de vida.

A janela de risco do certificado é no máximo igual ao seu tempo de vida, ou seja, o máximo intervalo de tempo em que o certificado se encontra válido e pode ser usado em caso de transvio. No entanto, é necessário ter em conta que a chave privada correspondente de um certificado se encontra dentro de um dispositivo criptográfico que bloqueia após um determinado número de tentativas falhadas, o que dificulta a sua utilização mal intencionada.

Uma das vantagens de certificados de curta duração são que torna mais simples

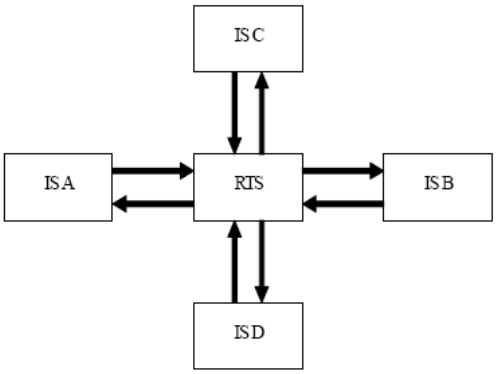

Figura 3.5: Modelo de Confiança Entre a RTS e as IS

fazer com que os profissionais façam pedidos de novos certificados e pares de chaves do que gerir a lista de certificados revogados.

Uma outra vertente tem a ver com a utilização do certificado para a RTS inferir sobre a autorização. Por exemplo, caso o cargo de um profissional seja alterado, seria necessário revogar o antigo certificado e emitir um novo. Com a utilização de certificados de curta duração este processo pode ser escolhido de forma a poder ter um grau de sincronismo adequado.

Para os certificados emitidos para Serviços/Servidores não são aconselháveis intervalos de validade curtos, porque como estão em ambiente controlado, o risco de comprometimento é muito menor. Caso estes certificados fossem de curta duração, havia o risco deles expirarem e os serviços ou servidores ficarem inacessíveis. Isto implica que para este tipo de certificados seja necessário gerir uma CRL.

Modelo de Confiança

A Figura 3.5 apresenta o modelo de confiança da RTS. Neste modelo, cada IS é responsável pela emissão e gestão dos certificados para as suas entidades. É assim que são definidas as instituições de origem da entidade possuidora de um certificado.

Para o estabelecimento de confiança mútua entre cada uma das IS e a RTS, e possibilitar que os certificados emitidos por cada uma das IS sejam válidos na RTS, e vice-versa, constitui-se uma estrela de certificações cruzadas na qual a RTS ocupa a posição central e as extremidades são ocupadas pelas Instituições de Saúde participantes na RTS. A certificação cruzada é feita individualmente com

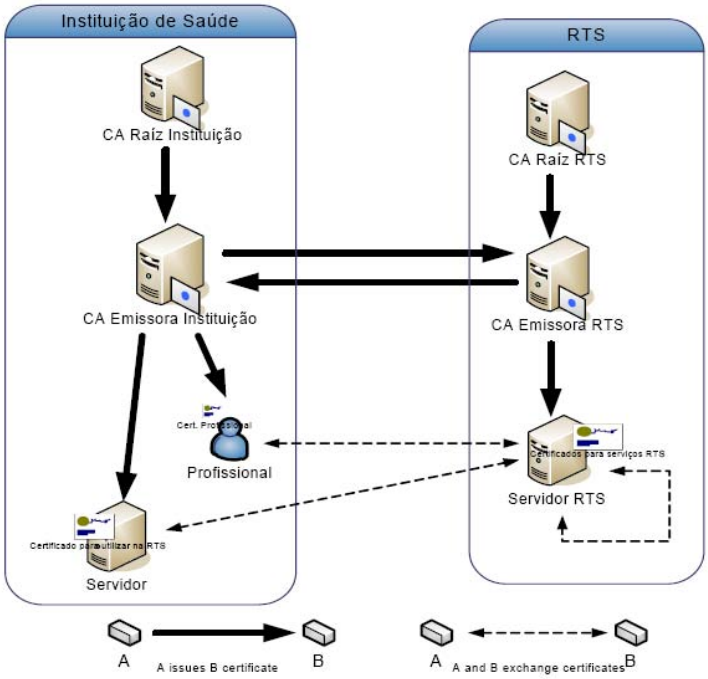

Figura 3.6: Arquitectura Interna das EC

cada IS e consiste na emissão de um certificado pela RTS para a IS e na emissão de um certificado da IS para a RTS.

Cada IS é responsável pela emissão dos certificados para as suas entidades. Caso alguma IS já possua alguma PKI em funcionamento preconiza-se a sua reutilização para a emissão de certificados para a RTS. Para as restantes IS sugere-se a implementação de uma hierarquia de ECs com pelo menos dois níveis tal como mostra a Figura 3.6.

A certificação cruzada para o estabelecimento de confiança entre a RTS e a IS é feita entre as EC Emissoras da RTS e da IS, que assinam os certificados uma da outra.

As razões para efectuar a certificação cruzada ao nível das EC Emissoras e não da EC raiz são limitar o âmbito da confiança entre as entidades, que se pretende apenas para os certificados a usar na RTS, e limitar danos em caso de quebra de confiança na outra instituição, uma vez que a publicação da CRL da EC Emissora é mais frequente que a da EC raiz, que está *offline*.

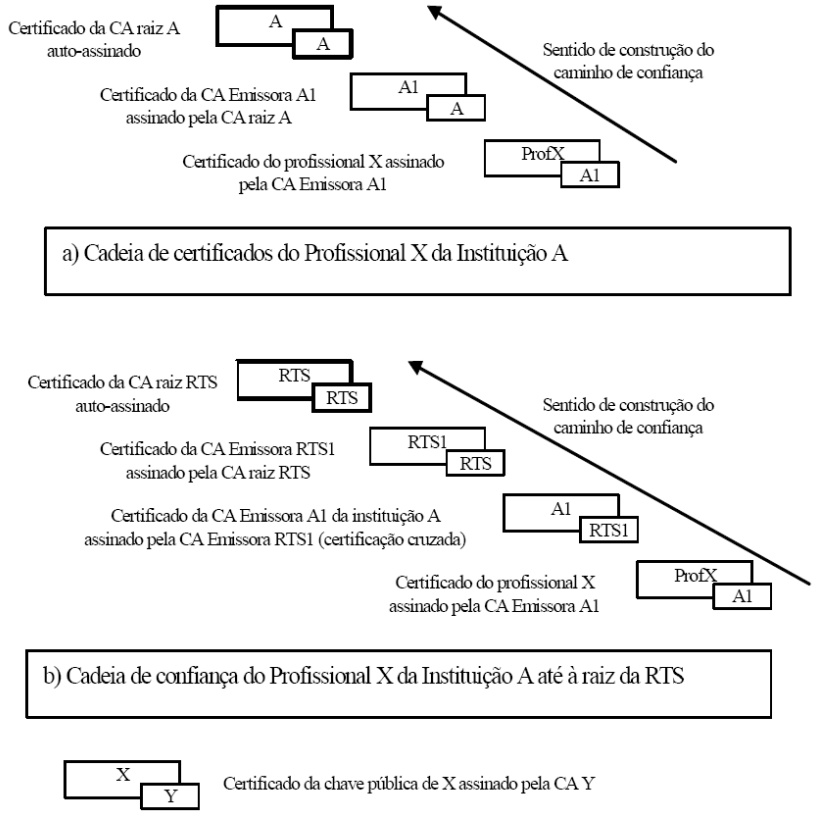

Figura 3.7: Construção de Cadeia de Confiança

Validação dos Certificados

Quando se inicia uma interacção na RTS, as entidades devem-se autenticar através da troca dos seus respectivos certificados digitais. Cada entidade deve validar o certificado da outra e proceder com a comunicação quando cada parte concluir que a outra é de confiança.

Uma identidade deve fazer as validações de certificados tendo como única âncora, ou raiz de confiança o certificado raiz da sua própria instituição. Para isso vai tentar construir o caminho ou cadeia de confiança que liga a entidade emissora do certificado a validar até ao certificado raiz da sua instituição. A validação do certificado recebido implica a validação de todos os certificados da cadeia de confiança constituída .A Figura 3.7 apresenta um exemplo da validação de um certificado e respectiva cadeia de confiança.

O problema com as certificações cruzadas é a dificuldade em obter de forma automática todos os certificados da cadeia. É recomendável que todas as enti-

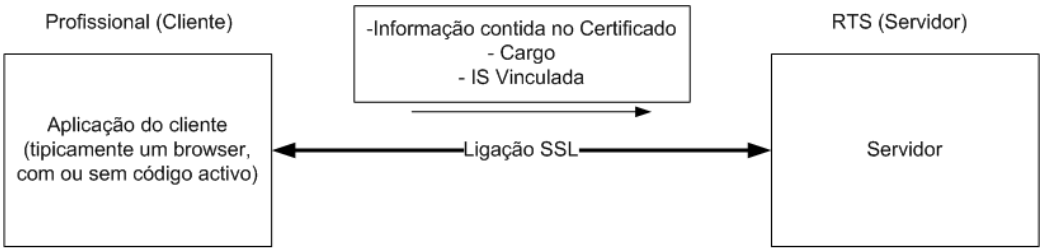

Figura 3.8: Comunicação Segura Entre o Profissional e a RTS

dades que participam na RTS estejam na posse dos certificados necessários para a construção das cadeias de confiança dos certificados da IS com as quais interagem. Isto implica que um Profissional terá que transportar no seu *smart card* o seu certificado, os certificados da sua cadeia hierárquica, e o certificado da certificação cruzada em que a EC Emissora da sua IS assina a chave pública da EC Emissora da RTS.

No caso dos Portais e restantes servidores da RTS que necessitem de validar certificados emitidos pelas IS, têm de possuir todos os certificados cruzados emitidos pela RTS para as várias IS, além dos certificados da cadeia hierárquica da RTS.

3.3.3 Adaptaçao da Arquitectura de Autenticação para o Uso do Cartão de Cidadão

É preciso encontrar uma maneira de introduzir ou associar as credenciais de um determinado Profissional a um cartão. Essas credenciais precisam de indicar o cargo do Profissional, a que IS ele está vinculado, e o período de validade. Também deve ser garantido que essas credenciais são genuínas, e apenas válidas para o possuidor daquele cartão.

Mesmo que as credenciais não estejam contidas no cartão, estas devem ser associadas a ele de alguma forma, de maneira a que o cartão sirva como prova de posse complementar às credenciais.

Entre o cliente e o servidor deve ser criada uma ligação SSL, como apresenta a Figura 3.8. Do lado do cliente deve correr uma aplicação que lê o Cartão de Cidadão, para autenticar o utilizador com base nos dados contidos no cartão, e ao mesmo tempo, que valide as credenciais desse utilizador, certificando-se que estas apenas são válidas na presença daquele cartão.

A informação obrigatória nas credenciais são o Cargo desempenhado pelo Profissional e a Instituição de Saúde ao qual está vinculado. Essa informação tem que chegar ao servidor da RTS de uma forma segura. A emissão dessas credenciais fica a cargo das Instituições de Saúde.

Uso do CC como *Smart Card* de Autenticação dos Profissionais

O Cartão de Cidadão já foi descrito no capítulo do Contexto. Nesta secção são mencionados os principais problemas que impedem que o Cartão de Cidadão seja utilizado sem problemas na arquitectura definida.

A adaptação da arquitectura de *smart cards* para o Cartão de Cidadão requer alguns cuidados. O Cartão de Cidadão não é um *smart card* normal, devido ao facto de estar limitado, sendo impossível a geração de novos certificados no seu interior.

Impossibilidade de Adicionar Novas Credenciais

O Cartão de Cidadão não permite criar novos certificados. Com *smart cards* o que se fazia era gerar no cartão um certificado para a RTS e outro para a IS, que agora não é possível. É possível verificar a autenticidade de uma pessoa mas não é possível saber qual o cargo desempenhado por essa pessoa na IS a que pertence pois a informação relativa à IS a que está vinculado e ao seu cargo não se encontram em nenhum certificado contido no cartão.

Ausência de Cargos

Sem a informação sobre o cargo executado por um determinado Profissional não é possível fornecer as permissões adequadas para que o Profissional possa aceder à área da sua especialidade, pois esta não é conhecida. O Cartão de Cidadão não possui meios de definir a informação sobre o cargo do Profissional.

Capítulo 4

Estudo da Solução com Código Activo

Neste capítulo são estudadas várias formas de colocar no *browser* dos profissionais capacidades de extensão do Cartão de Cidadão usando código activo (i.e. contido nos conteúdos enviados da RTS para o *browser*). As extensões visam fundamentalmente a obtenção de forma fidedigna na função profissional do cliente ao longo de uma sessão com a RTS mantendo parte da arquitectura de distribuição e validação de credenciais igual à do H. Gomes.

A Figura 4.1 apresenta uma solução com código activo.

Uma solução de código activo tem a vantagem de poder ser executada em *browsers* que não permitem o uso de módulos de segurança, e nesse caso o código activo trata da implementação dos mecanismos de segurança entre o cliente e o servidor, permitindo um maior controlo sobre o nível de segurança que se deseja implementar.

A única área do Cartão de Cidadão onde é possível escrever é o Bloco de Notas. É por essa razão que o acesso a esta área é tão importante no uso de soluções de código activo. Uma alternativa ao Bloco de Notas seria uma base de dados online que fizesse o mapeamento entre a identificação do cidadão e a informação extra relativa a ele, mas isso iria trazer outros custos, como por exemplo recursos de rede e definição de permissões para essa base de dados.

A abordagem escolhida foi o uso do Bloco de Notas para conter a informação sobre as credenciais. Não é uma imposição que elas estejam contidas no cartão, podem estar noutro tipo de suporte de média local ou até mesmo remoto, desde que apenas sejam válidas na presença física do Cartão de Cidadão, e essa validação

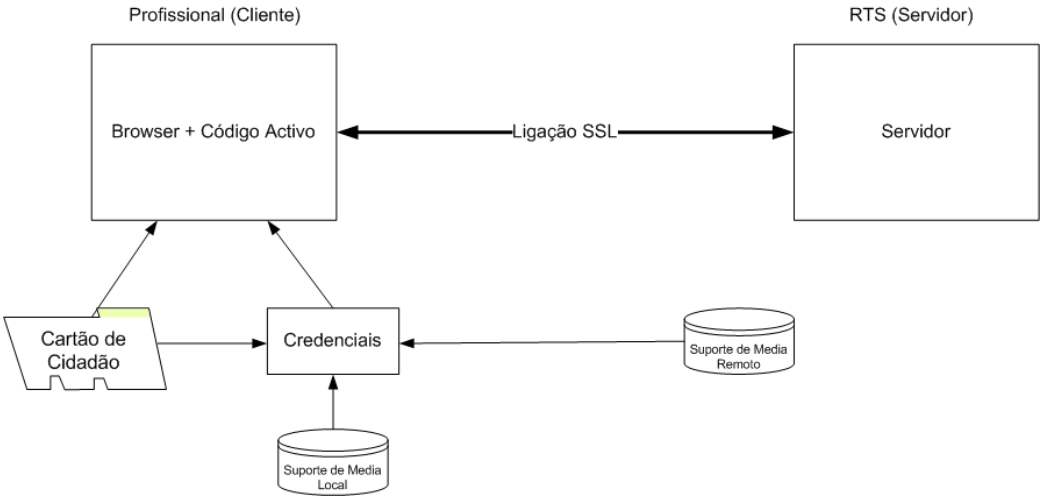

Figura 4.1: Solução com Código Activo

seja feita por código activo.

Considerou-se portanto, que para aumentar a mobilidade dos profissionais, as credenciais extra manipuladas por código activo seriam guardadas no Bloco de Notas do Cartão de Cidadão de cada profissional, e não num outro repositório avulso qualquer.

A existência de um mecanismo que vincule as credenciais de um Profissional ao seu cartão de cidadão é crítico. A capacidade de usar a biblioteca eID Lib é importante para a verificação da autenticidade das credenciais. As credenciais emitidas por uma Instituição de Saúde devem conter informações relacionadas com o certificado de autenticação do cidadão para impedir que sejam copiadas e utilizadas por por alguem não autorizado. Para aceder à informação contida no certificado de autenticação do cidadão é necessário recorrer à biblioteca eID Lib ou à PKCS#11.

A biblioteca eID Lib possui funções CVC que permitem a verificação da informação pessoa de uma maneira segura, impedindo que os dados sejam forjados, pois foram assinados [3]. Esta API também permite o acesso aos certificados X.509 contidos no CC. Alem disso, esta biblioteca é necessária para aceder ao Bloco de Notas, onde é colocada a informação sobre as credenciais do Profissional.

Caso as credenciais não sejam guardadas no Cartão de Cidadão, por falta de espaço ou por uma questão de facilidade na gestão das credenciais, mesmo assim elas devem estar vinculadas a um cartão. A maneira de verificar esse vínculo passa

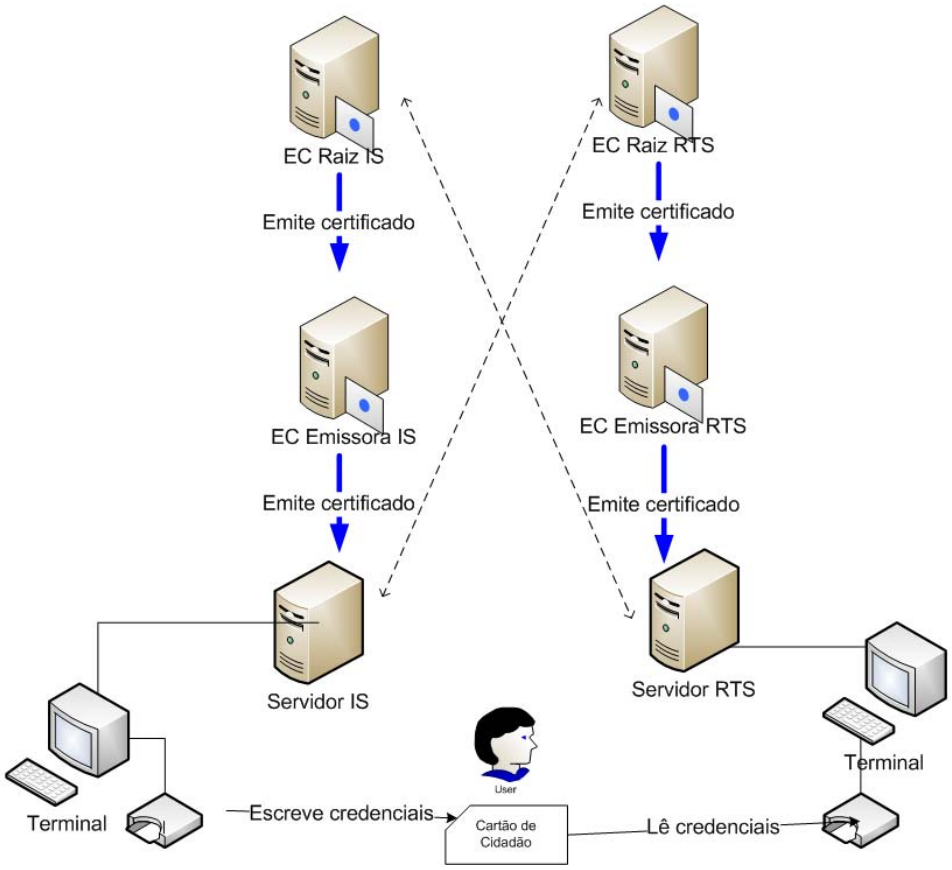

Figura 4.2: Relação entre a RTS e uma IS

pelo código activo.

Assume-se à partida que as entidades RTS e IS fazem parte de uma infra-estrutura de chaves públicas e que existe uma relação de confiança entre elas. Ambas as entidades também devem confiar na PKI do Cartão de Cidadão.

A Figura 4.2 descreve a relação entre uma IS e a RTS. As linhas a tracejado simbolizam relações de confiança. Uma instituição de saúde emite as credenciais a um profissional, que mais tarde ele usa para se autenticar na RTS.

Para aceder ao portal da RTS a primeira coisa a ser feita é a validação do certificado do Cartão de Cidadão do profissional, feito pelo *browser*, sem código activo ainda, e de seguida a validação das suas credenciais, para obter a informação do cargo do profissional em causa, através de código activo.

4.1 Modo de Funcionamento das Credenciais

Para lidar com a questão das credenciais emitidas pela RTS aos profissionais deve-se usar uma tecnologia que permita a emissão de credenciais por uma entidade e a sua validação perante outra entidade, como por exemplo a norma SAML (*Security Assertation Markup Language*). Esta norma é baseada em XML e a sua finalidade é fornecer dados sobre autenticação e autorização entre um fornecedor de identidade e um fornecedor de serviços.

SAML foi desenvolvido pelo comité técnico dos serviços de segurança da OASIS (*Organization for the Advancement of Structured Information Standards*). É uma *framework* baseado em XML que serve para comunicar a autenticação, título e informação adicional de um utilizador. SAML permite que entidades empresariais emitam assertações relativas a entidade, atributos e títulos de um sujeito (uma entidade que normalmente é um utilizador humano) a outras identidades, tais como uma empresa parceira ou outra aplicação empresarial. SAML é um protocolo flexível e extensível concebido para ser usado (e personalizado caso necessário) por outras normas.

Algumas das vantagens do SAML são que abstrai a *framework* de segurança da arquitecturas das plataformas e implementações de fornecedores particulares, não necessita que a informação do utilizador seja mantida e sincronizada entre directorias, permite que os utilizadores façam logon numa entidade provedora e que depois estes se autentiquem perante provedores de serviços sem autenticação adicional e reduz custos administrativos aos provedores de serviços.

No caso de logon único (*Single Sign-On*), um utilizador autentica-se perante um Web site e depois sem autenticação adicional, está apto a aceder a recursos personalizados por outro Web site. SAML activa o Web SSO através da comunicação de uma assertação de autenticação do primeiro site para o segundo, que se confiar no primeiro site pode escolher fazer o logon do utilizador como se este se tivesse autenticado directamente. O modelo é descrito na figura seguinte. Um utilizador autentica-se perante uma entidade que é reconhecida pela segunda identidade e lhe dá o acesso correspondente.

As assertações podem ser usadas com mensagens SOAP para transportar informações sobre entidade e segurança entre actores em transacções de serviços Web. O Perfil de Testemunhos SAML (*SAML Token Profile*) da *OASIS WS-Security TC* especifica como as assertações SAML devem ser usadas para este efeito.

SAML é definido em termos de assertações, protocolos, ligações e perfis.

Assertações: Uma assertação é o pacote de informação que fornece uma ou

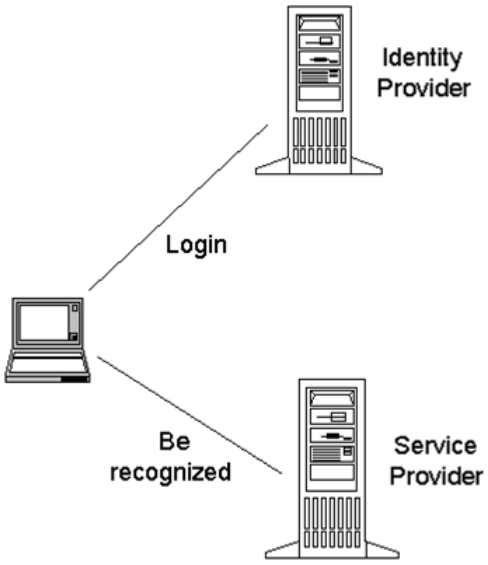

Figura 4.3: Modelo de autenticação SAML

mais definições declaradas por uma autoridade SAML. SAML define três tipos diferentes de declarações de assertação que podem ser criadas por uma autoridade SAML.

- **Autenticação (*Authentication*)**: O sujeito em questão foi autenticado de uma certa forma numa certa altura. Este tipo de declaração é tipicamente criado por uma autoridade SAML chamada provedor de identidade, que é responsável pela autenticação de utilizadores e manter o controlo de outras informações acerca deles.

- **Atributo (*Atribute*)**: O sujeito em questão é associado aos atributos fornecidos.

- **Decisão de Autorização (*Authorization Decision*)**: Um pedido para permitir que o sujeito em questão tenha acesso a um determinado recurso é permitido ou recusado.

4.1.1 Protocolos

SAML define um número de protocolos de pedido/resposta que permitem aos provedores de serviços:

- Pedir a uma autoridade SAML uma ou mais assertações.
- Pedir que um provedor de identificação autentique um utilizador e devolva a assertação correspondente.
- Pedir que um identificador de nome seja registado.
- Pedir que o uso de um identificador termine.
- Devolver uma mensagem de protocolo que foi pedida através de um artefacto.
- Pedir um logout quase simultâneo de todas as sessões relacionadas.
- Pedir um mapeamento do identificador de nomes.

4.1.2 Ligações (Bindings)

Mapeamentos de trocas de mensagens de pedido-resposta SAML em mensagens padrão ou protocolos de comunicação são denominados por *SAML Protocol Bindings*. Por exemplo, *SAML SOAP Binding* define como mensagens de protocolo SAML podem ser trocadas através de mensagens SOAP, enquanto *HTTP Redirect Binding* define como passar mensagens de protocolo através do redireccionamento de HTTP.

4.1.3 Perfis

Geralmente, um perfil em SAML define restrições ou extensões que facilitam o uso de SAML para uma determinada aplicação.

O perfil Web SSO (*Web SSO Profile*) define o uso do protocolo *SAML Authentication Request/Response* juntamente com diferentes combinações de *HTTP Redirect*, *HTTP POST*, *HTTP Artifact* e *SOAP* bindings.

Outro tipo de perfil SAML é um perfil de atributo. SAML define uma série de de perfis de atributo para fornecer regras especificas de interpretação de atributos em assertações de atributos SAML (*SAML attribute assertions*). Um exemplo é o perfil X500/LDAP, descrevendo como transportar atributos X.500/LDAP dentro de assertações de atributos SAML.

4.1.4 SAML e a RTS

A Figura 4.3 está de certa forma relacionada com a Figura 4.2. O fornecedor de identidade é a IS, e o fornecedor de serviços é a RTS. A mensagem que contem a assertação é passada através de um *token*, que é o Cartão de Cidadão.

4.2 Tecnologias Abordadas

Foi abordado um conjunto de tecnologias normalmente utilizadas em sites Web capazes de lidar com a biblioteca PKCS#11. Entre elas encontram-se *JavaScript*, *applets* Java e *ActiveX*. Como para o uso do código activo é necessário usar a biblioteca eID Lib, interessam tecnologias que permitam o uso de bibliotecas dinâmicas do sistema.

4.2.1 JavaScript

Esta linguagem por si só não suporta as funcionalidades necessárias para lidar com certificados nem com assinaturas digitais. Além disso não consegue ter acesso a certificados instalados no *browser* o que iria complicar a validação de cadeias de certificação, ou o acesso a chaveiros protegidos. O acesso ao sistema local de ficheiros também é bastante limitado, caso se quisessem adicionar funcionalidades para ler certificados a partir de ficheiros, fora do cartão, não seria possível ter acesso a eles.

O Firefox define um objecto em *JavaScript* para permitir que as paginas Web possam aceder a certos serviços criptográficos [1] Estes serviços estão balanceados entre a funcionalidade necessária de uma página e a protecção dos utilizadores de software maligno. Os serviços fornecidos permitem a geração de pedidos de certificados, importação de certificados de utilizadores, geração de números aleatórios e assinatura de texto.

Devido a razões de segurança, o *JavaScript* também não pode carregar bibliotecas dinâmicas (salvo raras excepções como é o caso de carregar um módulo PKCS#11, mas aí o *browser* impõe as limitações necessárias) para proteger os utilizadores. Desta forma o acesso ao bloco de notas do Cartão de Cidadão estaria impedido, impossibilitando que as credenciais dos profissionais fossem usadas.

[1] https://developer.mozilla.org/en/JavaScript_crypto

O uso de *JavaScript* também não é uniforme entre os *browsers*. Funções que são permitidas num *browser* podem não ser permitidas noutro, o que implica uma adaptação do código a cada tipo de *browser* que se quer utilizar. A qualquer altura pode ser lançada uma nova versão de um *browser* que considera que uma determinada função de *JavaScript* tem problemas de segurança e impedir o seu funcionamento, implicando uma nova restruturação no código de quem fornece a aplicação. Isso torna-se incómodo tanto para os utilizadores, que têm que esperar por uma actualização da aplicação, como para quem a desenvolve.

Em versões mais antigas dos *browsers*, era possível aceder ao sistema de ficheiros local usando *JavaScript* para ler ficheiros, mas com a evolução dos *browsers* e devido a questões de segurança, esses privilégios foram terminados. Uma maneira de resolver esta limitação usando a tecnologia *JavaScript* seria introduzir um campo num formulário em que o utilizador colocasse as credenciais manualmente, usando um método que validasse a autenticidade dessas mesmas credenciais embutido no código.

Apesar das limitações do *JavaScript*, esta tecnologia pode ser considerada (por exemplo como complemento a outra), pois mesmo sem conter todos os requisitos necessários para a concretização dos objectivos propostos, fornece bastantes mecanismos de suporte para o uso de certificados em *smart cards*.

Mesmo assim, não traz vantagens sobre o uso de um módulo de segurança carregado no *browser*, pois não permite que sejam chamadas funções de bibliotecas do sistema, por questões de segurança.

4.2.2 Java Applets

As *applets* java têm a vantagem de funcionar em qualquer tipo de *browser*, independentemente do sistema operativo onde se encontram. O acesso ao sistema de ficheiros locais pode estar limitado nas definições de segurança do JRE (*Java Runtime Enviorment*), mas isso pode ser redefinido. Continuam a depender de outros módulos para poder aceder aos certificados do cartão, como por exemplo o módulo *pteidpkcs11.dll* para aceder às funcionalidades da PKCS#11 e o módulo *pteidlibj.dll* para aceder ao bloco de notas do Cartão.

Em *applets* Java já é possível carregar bibliotecas dinâmicas, tal não acontecia com *JavaScript*. Neste caso, a política de segurança é definida pela Java Virtual Machine. À partida, qualquer *applet* assinada tem permissões para aceder ao sistema de ficheiros local se o *browser* confiar no certificado que assinou a *applet*. Se a *applet* for assinada por um certificado no qual o *browser* não confia, como por

exemplo, um certificado auto assinado, é perguntado ao utilizador se este aceita confiar na entidade que fornece aquela aplicação.

Uma *applet* em java pode usar qualquer biblioteca definida em Java. A linguagem Java fornece bibliotecas que simplificam o uso de módulos PKCS#11 permitindo efectuar o acesso à informação contida num *smart card* de uma forma abstracta à biblioteca PKCS#11 usando *wrapping*. O módulo PKCS#11 pode ser carregado pela *applet* e depois podem ser usadas funções contidas na *Java Cryptography Architecture* (JCA).

O objectivo da JCA é fornecer mecanismos de assinatura de documentos, verificação de assinaturas digitais e manipulação de certificados. A JCA foi criada para permitir diferentes implementações de vários serviços de diferentes criadores de software.

São fornecidas classes e interfaces para lidar com chaves públicas e privadas, certificados digitais, assinaturas de mensagens, verificação de assinaturas, acesso a chaveiros protegidos e outros mecanismos. Estas classes e interfaces são fornecidas pelos pacotes *java.security* e *java.security.cert*.

Uma *applet* Java funciona em qualquer sistema operativo que possua o *Java Runtime Enviorment*. Além disso, para correr as *applets* em *browsers* é necessário ter um *plug-in* instalado. Em alternativa, existe a possibilidade de correr a *applet* fora de um *browser*, usando a tecnologia JNLP. Para isso é criado um ficheiro JNLP que pode ser descarregado, e contem o caminho para a localização da *applet* e as bibliotecas do qual a *applet* depende, no seu código.

Ao executar um ficheiro JNLP, é lançada uma aplicação fornecida com o JRE chamada *Java Web Start*. Esta tecnologia permite correr uma *applet* fora de um *browser* como se fosse uma aplicação que estivesse a correr na maquina local.

Tem a vantagem de ser independente do *browser* ou do sistema operativo e tem a capacidade de aceder ao sistema local de ficheiros e de carregar bibliotecas dinâmicas do sistema ou exteriores ao sistema, se estiver assinado por uma entidade de confiança. O facto de um *browser* ser incapaz de lidar com a PKCS#11, como é o caso do Opera (até à versão 10 não suporta a instalação de módulos PKCS#11), não impede que a *applet* a use, devido à sua independência.

As desvantagens são que necessita do JRE instalado, em certos casos pode ser necessário redefinir permissões para a sua execução, e se tiver que carregar muitos ficheiros que existem no servidor pode consumir algum tempo sempre que é feito o download desses ficheiros.

A sua aplicação na RTS iria implicar uma nova implementação da interface

dos profissionais adaptada à *applet*. Iria implicar também a existência de código do lado do servidor para que as credenciais dos profissionais podessem ser exploradas, o que tornaria a sua implementação de alguma forma complexa.

4.2.3 ActiveX

O *ActiveX* à partida é uma solução válida, mas está limitada ao Internet Explorer, consequentemente, a sistemas Microsoft Windows. Apesar de outros *browsers* suportarem controles *ActiveX*, esta tecnologia só funciona em ambiente Microsoft Windows. Foi por esta razão que esta solução foi também colocada de parte.

4.2.4 Macromedia Flash

Esta tecnologia não permite o uso de certificados ou assinaturas digitais, e também não pode aceder ao sistema de ficheiros local ou ao Bloco de Notas do Cartão de Cidadão.

4.3 Avaliação

Dadas as várias tecnologias analisadas, e os requisitos para o sistema sendo a simplicidade entre a interacção do profissional com o portal, independentemente do sistema operativo ou do *browser*, a melhor solução de código activo seria o uso da *applet*, pois permite uma maior flexibilidade entre as plataformas e a manipulação da informação contida no Cartão de Cidadão.

No entanto a sua implementação não foi concretizada, porque esta solução implica código activo do lado do cliente e do lado do servidor. Mesmo assim, é uma solução com bastante potencialidade.

Capítulo 5

Estudo da Solução sem Código Activo

Neste capítulo estuda-se um método que coloca um certificado contido no Bloco de Notas do Cartão de Cidadão no *browser* do profissional usando um *wrapper*. Esse certificado contém extensões relativas ao cargo do profissional, definidas da mesma maneira que na arquitectura concebida pelo H. Gomes, ou seja, com um OID (*Object ID*) relativo ao cargo desempenhado por um determinado profissional.

Na Figura 5.1 é descrito o modo como é feita a emissão de um certificado por uma Instituição de Saúde. Esse certificado fica guardado no Bloco de Notas do Cartão de Cidadão e usa o mesmo par de chaves do certificado de autenticação do Cartão de Cidadão.

5.1 Wrapping do Cartão de Cidadão

Este *wrapper* funciona como um *proxy* entre o *browser* e o módulo PKCS#11 fornecido no software do *middleware* do Cartão de Cidadão.

Um dos seus objectivos iniciais era construir um *Log* para perceber quais eram as funções do módulo PKCS#11 chamadas pelo *browser* e a partir daí se perceber como é que o *browser* lida com um módulo PKCS#11.

O principal objectivo é acrescentar a funcionalidade para que possa ser lido um certificado contido no Bloco de Notas do Cartão de Cidadão que usa o par de chaves de autenticação já existente no cartão, criando a ilusão ao *browser* de que

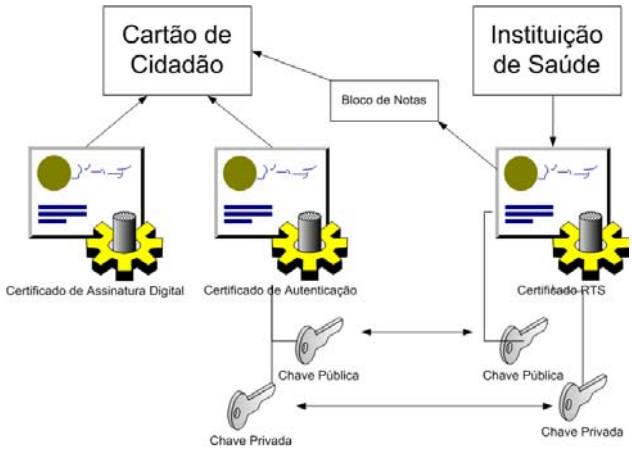

Figura 5.1: Arquitectura

se trata de outro par de chaves.

Depois de perceber como é que o *browser* lida com a PKCS#11, procedeu-se à implementação dos métodos necessários para adicionar o certificado contido no bloco de notas do Cartão de Cidadão no browser. Na maior parte dos casos, sempre que uma função do *wrapper* é invocada, é chamada a função correspondente da PKCS#11 contida no modulo fornecido no *middleware* do Cartão de Cidadão. Isto nem sempre acontece no caso das funções *C_GetFunctionList*, *C_GetAttributeValue* e *C_FindObjectsInit*, como vai ser explicado posteriormente.

5.1.1 Modo de Funcionamento

O wrapper actua entre o *browser* e o módulo fornecido no *middleware* do Cartão de Cidadão, como se pode ver na Fig. 5.2 fazendo o encaminhamento das funções que o *browser* pede ao módulo da PKCS#11.

A primeira função executada pelo *browser* é a *C_GetFunctionList*, que faz o pedido ao módulo da lista de funções e guarda os endereços das funções nele contidas. Esta função não pode ser mapeada directamente, porque caso o fosse, iria conter os endereços das funções do módulo *pteidpkcs11.dll*, e o que se quer é que os endereços obtidos pelo *browser* sejam os das funções do *wrapper*. Só assim se pode enviar a lista das funções chamadas pelo *browser* para um ficheiro.

O *Log* consiste numa lista de todas as funções invocadas pelo *browser* exactamente pela mesma ordem. Foi através da análise desta lista que se percebeu quais eram os pedidos feitos pelo *browser* ao módulo da PKCS#11. Depois de perce-

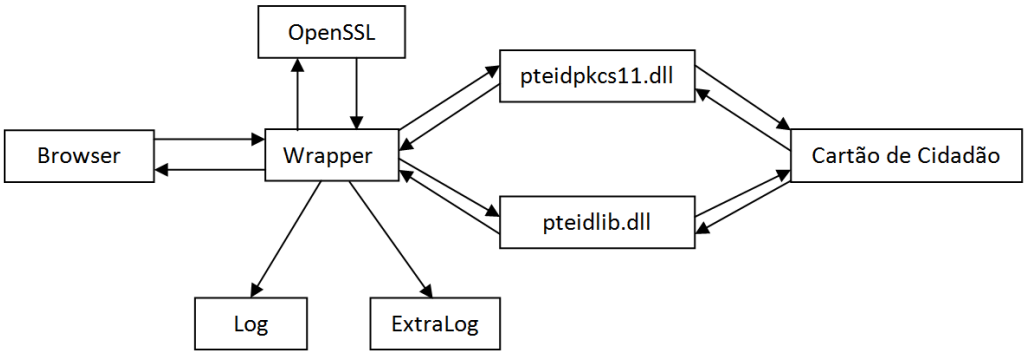

Figura 5.2: Interacção Entre o Browser e o Cartão de Cidadão usando o Wrapper

ber essa ordem e o que significa cada pedido facilmente se "engana o *browser*" fornecendo-lhe a informação do novo certificado que se quer adicionar.

Um método mais eficiente seria em vez do uso do *wrapper*, apenas um módulo que tivesse todas as funcionalidades da PKCS#11 e além disso permitisse a leitura de certificados do Bloco de Notas. Para a implementação de um módulo desses seria necessária a colaboração da entidade responsável pela produção do software do *middleware* do Cartão de Cidadão. Nessa solução podiam ser eliminadas as dependências da biblioteca dinâmica do OpenSSL.

5.1.2 Descrição dos Componentes Utilizados

O *wrapper* usa a API do OpenSSL, para poder extrair certos campos do certificado que está no formato PEM usando funções específicas relativas ao que é pedido. Por exemplo, para passar um certificado no formato PEM para uma estrutura de certificado X509 [12], é possível usar a função *PEM_read_bio_X509* fornecida pela API do OpenSSL. A dependência da biblioteca dinâmica do OpenSSL não é a melhor ideia, e deveria ter sido evitada, o custo seria reflectido na implementação do *wrapper*, mas a sua instalação seria mais simples, diminuindo a lista de dependências.

Por uma questão de facilidade na implementação, as razões principais pelas quais foram usadas funções do OpenSSL foram a existência de estruturas de certificados X.509, e funções úteis que manipulam esses dados. O uso do formato PEM foi ponderado, porque também traz alguns custos e alguns benefícios. Um dos custos é o espaço que ocupa em relação ao espaço ocupado por um certificado no formato DER.

Um certificado no formato PEM ocupa cerca de quatro terços do tamanho de um certificado no formato DER. De qualquer maneira seria quase impossível incluir dois certificados em formato DER incluindo algo que permitisse ver a separação entre os dois no espaço disponível do cartão.

Um utilizador pode ver onde começa e termina um certificado no formato PEM e facilmente copiá-lo para dentro ou para fora do Bloco de Notas usando apenas a aplicação fornecida pelo software do *middleware* do Cartão de Cidadão. Caso seja um profissional que pertence a várias Instituições de Saúde, e possua vários certificados cada um correspondente a uma Instituição, pode geri-los apenas com um ficheiro de texto, copiando para o cartão o certificado que precisa quando se quer autenticar na a RTS.

O *wrapper* também usa a biblioteca dinâmica *pteidlib.dll* para aceder ao bloco de notas do Cartão de Cidadão e extrair o certificado lá existente.

Como não podia deixar de ser, o *wrapper* também usa a API da PKCS#11 que faz a interacção com o *browser*, e com o módulo fornecido pelo Cartão de Cidadão, para poder manipular as estruturas dos dados pedidos pelo *browser* que são modificados no *wrapper*, para "enganar" o *browser*.

5.1.3 O Módulo PKCS#11

A PKCS#11 é composta por um conjunto de 68 funções definidas conforme a norma em [4].

Dessas funções apenas 31 foram implementadas no *middleware* do Cartão de Cidadão, descrito na secção 2.4. A função *C_CreateObjects* não foi implementada, o que limitou o uso dos novos objectos ao nível do *wrapper*. Uma solução melhor seria a criação de um novo par de chaves no cartão e o uso desse par de chaves no certificado adicionado. Como tal não é possível, os novos objectos são simulados pelo *wrapper*.

Quatro das principais funções alteradas no *wrapper* para fazer a interacção com o *browser* são a *C_GetAttributeValue*, que obtém um valor para um determinado atributo de um determinado objecto. As outras três funções são a *C_FindObjectsInit* que inicia uma pesquisa por um objecto de um determinado tipo, é seguida da função *C_Findobjects* que devolve o *handle* para o objecto caso ele seja encontrado, assim como o número de objectos encontrados, e assim que a busca por novos objectos termina é chamada a função *C_FindObjectsFinal*.

Figura 5.3: Caixa de diálogo de inserção de PIN

5.1.4 Interacção entre o *browser* Firefox e a PKCS#11

Nem todas as funções chamadas pelo *browser* devolvem o código CKR_OK (0). São elas a *C_WaitForSlotEvent*, que devolve o código 8, que corresponde ao erro CKR_NO_EVENT e *C_GetAttributeValue*, que devolve o código 18 e corresponde ao erro CKR_ATTRIBUTE_TYPE_INVALID. Estes códigos, apesar de não serem 0, não significam que tenha ocorrido algum erro.

No caso do CKR_NO_EVENT, significa que nenhum cartão foi retirado ou inserido durante a verificação. No caso do CKR_ATTRIBUTE_TYPE_INVALID, acontece porque foi declarado um atributo que não é suportado pelo módulo ou o objecto não possui o tal atributo.

A função *C_Login* não aparece porque se pode fazer um *C_Sign* sem se fazer um Login, a biblioteca PKCS#11 é que decide o que faz, se rejeita ou se aceita o PIN através de uma interface própria (Figura 5.3). O *browser* arrisca, não faz *Login*, e a biblioteca apresenta a caixa de diálogo do PIN.

Para contextualizar, é resumidamente descrita a interface da função *C_GetAttributeValue*. Esta leva como parâmetros de entrada um handle para a sessão (*hSession*), um handle para o objecto (*hObject*), um ponteiro para um *template* com tipo de atributo obtido (*pTemplate*), e o número de atributos no *template* (*ulCount*). Esse *template* é composto por um ou mais atributos. Cada atributo tem um tipo definido, um valor, e o tamanho ocupado pelo valor.

A função *CC_GetAttributeValue* é sempre executada duas vezes seguidas pelo *browser*, sendo que na primeira apenas é devolvido o tamanho que o valor do atributo ocupa, para que o *browser* reserve o espaço necessário em memória. Por isso, na primeira invocação, o valor do campo *pValue* do atributo vai com o valor NULL e na segunda vez já vai com o valor da zona de memória reservada para

0	CKA_CLASS:	CK_OBJECT_CLASS (CK_ULONG)
1	CKA_TOKEN:	CK_BBOOL
2	CKA_PRIVATE:	CK_BBOOL
3	CKA_LABEL:	RFC2279 String
17	CKA_VALUE:	Byte Array
128	CKA_CERTIFICATE_TYPE:	CK_CERTIFICATE_TYPE (CK_ULONG)
129	CKA_ISSUER:	Byte Array
130	CKA_SERIAL_NUMBER:	Byte Array
256	CKA_KEY_TYPE:	CK_KEY_TYPE (CK_ULONG)
257	CKA_SUBJECT:	Byte Array
258	CKA_ID:	Byte Array
288	CKA_MODULUS:	Big Integer

Tabela 5.1: Alguns Atributos da Cryptoki

guardar os dados do atributo.

Os vários tipos de atributos detectados no *log* são descritos na Tabela 5.1.4:

A 1ª coluna contém o código do atributo em formato decimal, a 2ª coluna contém a designação do tipo de atributo e a terceira coluna contém o tipo de dados.

5.2 Exploração de Tokens via PKCS#11 com o Firefox

O *browser* Firefox possui a funcionalidade de adicionar módulos de segurança. É assim que é adicionado o modulo do *wrapper* ao *browser*. Para adicionar um módulo ao Firefox deve-se abrir o menu "Ferramentas", seleccionar "Opções", a aba de "Cifra" e "Dispositivos de Segurança", Figura 5.4. Depois seleciona-se o botão "Carregar", Figura 5.5, e aparece uma nova caixa de diálogo, Figura 5.6, onde se escolhe o nome do módulo e o caminho para o ficheiro.

O Firefox também permite importar certificados PKCS#12, cuja chave privada é guardada num ficheiro. Como é impossível extrair a chave privada contida no Cartão de Cidadão para um ficheiro, essa opção ficou logo excluida, e se se optasse por usar um par de chaves gerado apenas com o propósito de autenticar um profissional na RTS, o Cartão de Cidadão deixava de ser preciso, podendo o

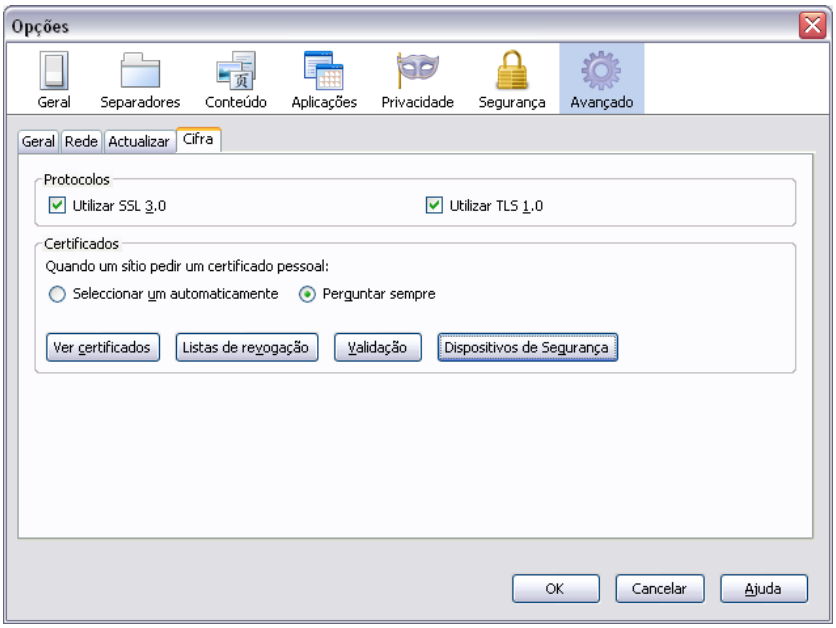

Figura 5.4: Adicionar um Módulo - Parte 1

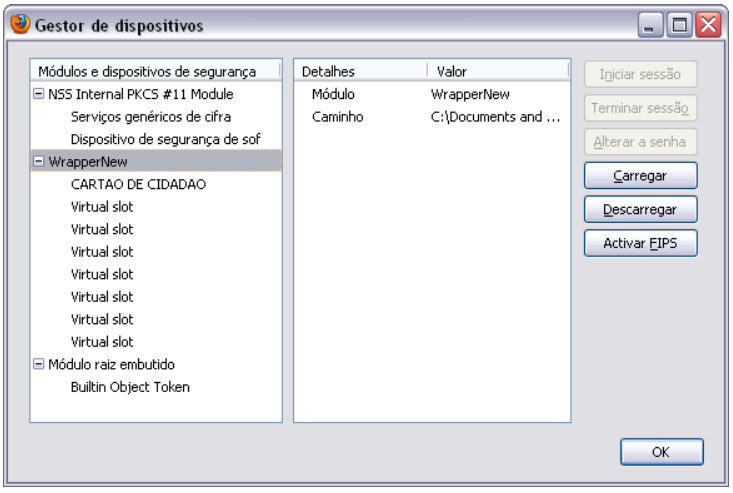

Figura 5.5: Adicionar um Módulo - Parte 2

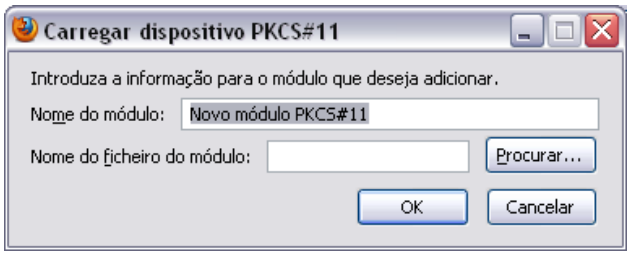

Figura 5.6: Adicionar um Módulo - Parte 3

certificado ser guardado num *Token*.

Mesmo que se quisesse aproveitar o Cartão de Cidadão apenas para guardar a informação do certificado e a sua chave privada isso também não seria possível devido ao escasso espaço nele disponível. Foram estas as razões principais que levaram por optar pela solução do uso da PKCS#11 que consiste em fornecer ao *browser* a informação sobre um novo certificado existente no cartão da forma a seguir descrita.

O processo utilizado pelo Firefox para procurar certificados em *tokens* que usam a PKCS#11 muito resumidamente consiste no seguinte:

- Procurar um token disponível.
- Iniciar uma sessão nesse token, caso tenha sido encontrado um.
- Procura por objectos do tipo "Certificado".
- Verifica o label dos certificados e se têm o atributo "token" definido como "TRUE".
- Obtém o valor do atributo "CKA_VALUE" de cada certificado.
- Obtém o valor de todos atributos de cada certificado.
- Verifica o valor do atributo "CKA_ID" de cada certificado e efectua uma pesquisa por objectos do tipo "Chave Privada" com o mesmo valor do atributo "CKA_ID".

Os objectos existentes no cartão são de 3 tipos diferentes, ou são do tipo "Private Key", ou do tipo "Certificate", ou do tipo "Public Key". Podem haver objectos de outros tipos, como por exemplo do tipo "Data", mas apenas estes descritos anteriormente na secção 2.4, na Tabela 2.4.3, existem no cartão.

CKA_CLASS:	A classe do objecto
CKA_TOKEN:	Se o objecto é token ou objecto de sessão
CKA_LABEL:	Contém o nome designado ao certificado
CKA_CERTIFICATE_TYPE:	De que tipo se trata
CKA_ID:	O ID do objecto
CKA_VALUE:	O valor do certificado (o seu conteúdo)
CKA_ISSUER:	A entidade emissora do certificado
CKA_SERIAL_NUMBER:	O número de série
CKA_SUBJECT:	Informação sobre o sujeito do certificado

Tabela 5.2: Atributos de um Certificado

Todos os objectos do tipo "Certificado" que têm um objecto do tipo "Chave Privada" associado são adicionados ao *browser* e é possível ver os seus atributos do seguinte modo:

Ferramentas → Opções → Avançado → Ver certificados → Os seus certificados

No entanto isto não significa que seja possível usá-los para efectuar uma ligação SSL. Se 2 certificados partilharem a mesma chave privada, ou seja, se o *handle* do objecto for o mesmo, apenas aparece um certificado disponível quando a escolha de certificados é solicitada pelo *browser*.

O Cartão de Cidadão possui alguns *object handles* definidos no módulo fornecido por ele, relativos aos certificados nele contidos, chaves privadas e chaves públicas correspondentes.

Tiveram que ser criados 2 objectos novos, um correspondente ao novo certificado, e outro correspondente à sua chave privada, que funcionam ao nível do *wrapper*.

Um objecto do tipo certificado deve incluir os atributos indicados na Tabela 5.2.

O atributo CKA_ID do certificado deve ter o mesmo valor que o atributo CKA_ID da chave privada correspondente para efeitos de pesquisa de objecto. Ora, caso seja detectada a pesquisa por um objecto do tipo PRIVATE KEY cujo valor de CKA_ID é igual ao valor de CKA_ID do novo certificado, pode-se devolver um *handle* para a nova chave privada, que não é mais do que a chave privada do certificado de autenticação mascarada.

Os novos objectos oram definidos como 101 para o Certificado e 102 para a chave privada. Na verdade, o único objecto mesmo criado é o certificado, porque

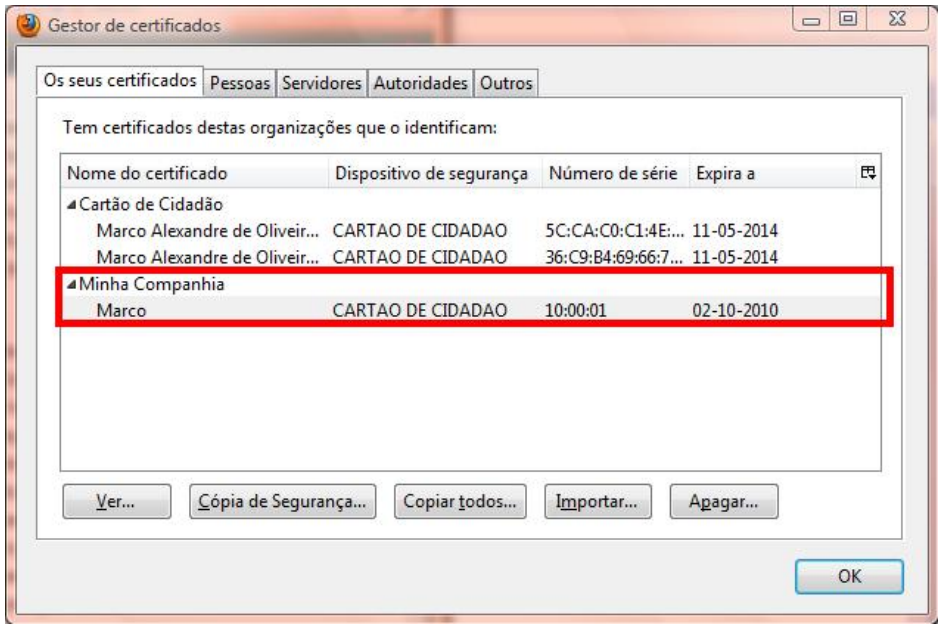

Figura 5.7: Gestor de Certificados do Firefox

a chave privada é a mesma, apenas teve que se definir um *handle* diferente para ela para que o *browser* detectasse uma chave privada diferente e permitisse que o novo certificado pudesse ser utilizado. Caso esse *handle* para a chave privada não fosse criado, o que acontecia era que se poderia ver que o novo certificado se encontrava instalado mas não era possível utilizá-lo para iniciar uma sessão segura com um servidor.

Na Figura 5.7 pode-se ver o novo certificado instalado no *browser* Firefox.

Agora é possível aceder ao portal e efectuar a autenticação usando o certificado da RTS. A Figura 5.8 mostra os dois certificados à escolha.

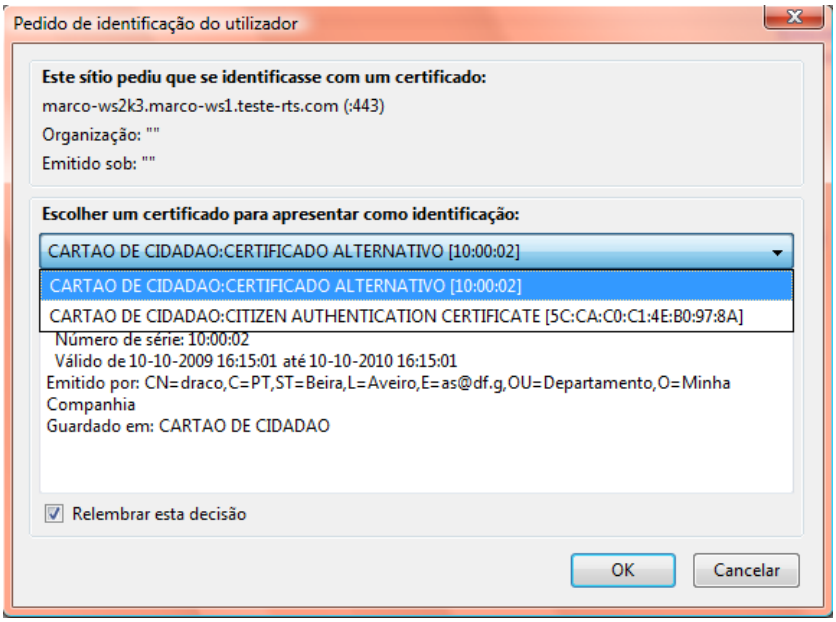

Figura 5.8: Selecção de Certificado

Capítulo 6

Arquitectura Proposta

Neste capítulo propõe-se uma arquitectura para a autenticação de profissionais que pertencem à RTS. Esta arquitectura baseia-se nos pressupostos definidos pelo H. Gomes, ou seja, na utilização de pares de chaves assimétricas e certificados digitais de chave pública enquadrados numa infra-estrutura de chave pública (*Public Key Infrastructure*, PKI). Ao longo do texto é feita uma comparação com o que tinha sido feito pelo H. Gomes bem como a sua adaptação para o uso do Cartão do Cidadão

6.1 Apresentação

Cada Instituição de Saúde (IS), é responsável pela emissão e gestão de certificados para as suas entidades - serviços/servidores e Profissionais. A RTS deve incluir os certificados das Entidades Certificadoras (EC) das IS emissoras na sua lista de entidades de confiança, permitindo que um Profissional que possua um certificado válido tenha acesso ao portal da RTS.

Em vez dos *smart cards* utilizados na arquitectura definida pelo H. Gomes, o transporte e armazenamento das credenciais dos profissionais é feito com o Cartão de Cidadão, usando um certificado emitido pela IS a que pertence o profissional, e que se serve do par de chaves do certificado de autenticação contido no cartão. Este cartão, tal como um *smart card* comum, permite um nível de autenticação forte (posse e senha), sendo necessária a introdução de um código PIN quando é feita a autenticação perante o portal da RTS.

Como o par de chaves utilizado na autenticação é o que vem com o certificado

do Cartão de Cidadão, a sua renovação está fora de questão, pois o cartão não permite a criação de novos pares de chaves. Sendo assim, não faz sentido usar 2 certificados, um para renovação de chaves e outro para a autenticação.

É usado um único certificado de autenticação emitido pela IS. Este certificado tem a função de autenticar um determinado Profissional na RTS, indicando o cargo desempenhado por esse mesmo Profissional. Deve ser usada uma CRL usada na validação dos certificados, pois estes terão um tempo de vida relativamente longo. A emissão teria que ser validada por um funcionário responsável, que iria verificar se um determinado funcionário está apto a executar o cargo pretendido no pedido de certificado.

Em alternativa, existe a opção de usar certificados de curta duração, eliminando a existência de CRLs, mas obrigando a que os profissionais façam uma renovação de certificados mais frequente, tal como acontecia na definição proposta pelo H. Gomes. Isso iria libertar a carga do sistema, mas traria outros custos, como por exemplo a existência de uma lista de roles disponíveis para cada Profissional mapeada ao certificado do Cartão de Cidadão do Profissional. A sua gestão caberia às IS. Isto não eliminaria a existência de um funcionário responsável pela gestão de permissões das funções dos Profissionais.

Tal como na arquitectura de definida pelo H. Gomes, no modelo de confiança utilizado, cada entidade tem como âncora de confiança a entidade certificadora de raiz da instituição a que está vinculada.

O Cartão de Cidadão não possui espaço disponível para guardar as cadeias de certificação. Estas devem ser obtidas online e instaladas manualmente no *browser* utilizado para aceder à RTS, para que a validação das cadeias de certificação seja realizada e consequentemente a autenticação do Profissional.

6.2 Certificados Digitais

O formato dos certificados digitais a utilizar na autenticação entre as entidades nos acessos à RTS obedece ao perfil de certificados X.509 para utilização na Internet, recomendado pelo IETF no RFC 3280 [13]

6.2.1 Entidades com Certificados

Os certificados digitais de chave pública utilizados, servem para autenticar as entidades que comunicam no âmbito RTS. Estas entidades podem ser de dois tipos,

Utilizadores ou Servidores/Serviços.

6.2.2 Tipos de Certificados

Os tipos de certificados para Servidores/Serviços têm como finalidade a autenticação. Os certificados dos Profissionais, além de também servirem para a autenticação, têm a função de fornecer ao Portal a informação sobre o cargo desempenhado pelo Profissional.

Irão haver dois tipos de certificados para autenticação no acesso à RTS, um tipo para os Profissionais e outro tipo para os Servidores/Serviços.

6.2.3 Formato dos Certificados

Os campos obrigatórios que devem ser definidos nos certificados são:

- **Sujeito (*Subject*)**: Contem o nome da entidade para quem o certificado foi emitido e da IS a que pertence. A informação é especificada usando etiquetas (tags), em que "CN" indica o nome do Profissional, "O" ou "DN" identificam a IS. Não convém usar mais do que as etiquetas necessárias para que o tamanho do certificado não exceda os 1000 bytes para os certificados dos Profissionais.

- **Emissor (*Issuer*)**: Identificação da EC que emitiu o certificado.

- **Validade**: Intervalo de tempo em que o certificado é válido.

- **Extensão EKU**: Aqui é definida a finalidade do certificado. No caso dos certificados dos Profissionais, este campo deve conter um OID (*Object Identifier*) de Autenticação de Cliente e outro de identificação do perfil do Profissional na IS a que está vinculado. No caso dos certificados de Servidores/Serviços, devem ser definidos os OIDs Autenticação de Cliente e Autenticação de Servidor.

- ***CRL Distribution Point* (CDP)**: Aqui é definido o endereço electrónico do local onde é emitida a lista de certificados revogados (CRL).

6.2.4 Gestão de Certificados

A gestão é feita por cada IS, que sendo instituições autónomas, devem fazer cada uma a própria gestão das listas de certificados revogados. Se um certificado for válido e não estiver na lista de certificados revogados emitida pela IS que emitiu esse certificado, então ele é aceite pela RTS.

Um profissional que esteja vinculado a mais do que uma Instituição de Saúde pode ter mais do que um certificado, mas o cartão apenas permite conter um certificado nele guardado, devido à limitação do espaço. Nesse caso, a gestão dos certificados deve ser feita pelos próprios Profissionais. Devem manter os seus certificados guardados noutro local, e sempre que quiserem usar um determinado certificado, podem usar a aplicação fornecida com o Cartão de Cidadão para o copiar para o bloco de notas. Sempre que é feita alguma alteração no Bloco de Notas é obrigatório introduzir o código PIN.

6.2.5 Emissão/Renovação

Em caso de perda do Cartão de Cidadão ou danos no chip, que impeçam o seu funcionamento, é necessário emitir um cartão novo, que vem com um novo par de chaves. Nesse caso a IS deve emitir um novo certificado com o novo par de chaves. Caso haja alguma alteração no perfil do Profissional, também é necessária a emissão de um novo certificado.

O pedido de certificado (CSR) tem que ser assinado com a chave privada do certificado de autenticação do Cartão de Cidadão do Profissional. Isso implica a presença do próprio, pois é necessário introduzir o código PIN.

O pedido de certificado pode ser feito perante a Instituição de Saúde, ou usando um mecanismo online, que pode ser concebido para efectuar esse pedido. No pedido o profissional deve introduzir os dados obrigatórios definidos em 6.2.3 e esperar que lhe seja enviado o certificado.

6.2.6 Armazenamento

O Profissional, depois de obter o seu certificado deve usar a aplicação fornecida com o Cartão de Cidadão para aceder ao Bloco de Notas e aí inserir o seu certificado.

Não cabe à RTS impor regras. A emissão dos certificados aos profissionais,

é feita pelas Instituições de Saúde, mas para que o Cartão do Cidadão possa ser usado como meio de autenticação, as IS devem cumprir alguns requisitos como por exemplo minimizar a informação contida no certificado para que este não exceda os 1000 bytes.

6.2.7 Cadeias de Certificação

Tanto a RTS como as IS têm as suas próprias cadeias de certificação. No entanto, existe uma relação de confiança entre a RTS e a IS.

Para a RTS, existe uma EC raiz que emite certificados para a(s) EC emissora(s) da RTS, que por sua vez emitem os certificados para o Portal e Serviços.

Para as IS, o processo é semelhante, existe uma IS raiz que emite certificados para a(s) EC emissora(s) das IS respectivas, e estas emitem certificados aos Profissionais que têm o direito de lhes aceder através do Portal da RTS.

O Portal da RTS deve confiar nas EC raiz das IS que colaboram com a RTS, para que a autenticação de cada Profissional pertencente à IS colaboradora seja validada.

As entidades da RTS e da IS que comunicam entre si devem ter na sua lista de EC de confiança as EC raiz da entidade com quem efectuam a ligação SSL.

Devido às limitações do Cartão de Cidadão, é impossível guardar no cartão a cadeia de certificação do certificado do Profissional. Por essa razão, a RTS e a IS devem fornecer os certificados da EC emissora e raiz da IS e da RTS, para que estes possam ser instalados no *browser*, pois caso verificação da cadeia de certificação não tenha sucesso, a autenticação falha.

A Figura 6.1 apresenta a hierarquia das Entidades Certificadoras da IS e da RTS respectivamente.

6.3 Geração de um Pedido de Certificado

Devido ao espaço limitado (1000 Bytes) e a impossibilidade de criar novos pares de chaves no cartão, que é uma desvantagem em relação aos *smart cards* normais, optou-se pela reutilização da chave privada de autenticação já nele existente.

O cartão possui dois pares de chaves, o de Autenticação e o de Assinatura [7].

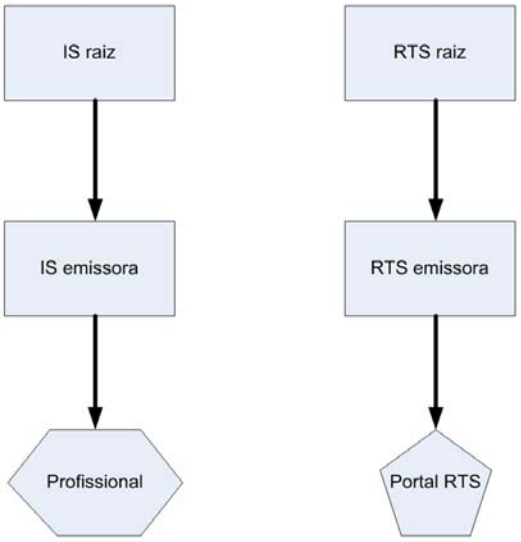

Figura 6.1: Cadeias de Certificação: IS - RTS

O par de chaves escolhido para ser reutilizado foi o de Autenticação.

Após uma modificação no código do OpenSSL, é possível emitir certificados que usam o par de chaves de autenticação do Cartão de Cidadão usando a ferramenta de criação de pedidos de certificados fornecida por esta biblioteca.

Define-se um ficheiro de configuração do OpenSSL que contem as extensões com a informação do cargo do profissional.

É preciso criar um par de chaves num ficheiro, cujo o único efeito é permitir que o OpenSSL reserve internamente memória para a chave pública no certificado criado, que depois é substituida pela chave pública do Cartão de Cidadão, sem que seja gerado um par novo no acto de criação, assumindo que o dono do certificado já tem a chave privada em seu poder, como é o caso.

É gerado um ficheiro no formato PEM que contém o pedido de certificado. Este pedido é enviado à Entidade Certificadora.

A entidade certificadora está configurada para poder emitir certificados com extensões consoante o cargo do funcionário que fez o pedido.

É executado o comando que processa a emissão do certificado requerido, e este é criado e fornecido ao profissional que o requisitou.

Esse certificado vai num ficheiro PEM, cujo conteúdo pode ser copiado para o bloco de notas do Cartão de Cidadão, usando a aplicação fornecida com o *mid-*

dleware do Cartão de Cidadão.

6.4 Processo de Autenticação

O processo de autenticação na prática apenas requer que um determinado profissional possua um certificado válido no Cartão de Cidadão. É importante referir que o processo de autenticação não é exclusivo apenas ao Cartão de Cidadão. É possível usar outro certificado contido num *smart card* ou *token* USB, desde que tenha sido emitido por uma IS colaboradora com a RTS.

Quando é feito o acesso ao portal, o *browser* pede para que seja escolhido um certificado, o Profissional escolhe o certificado adequado, e se este for válido, é iniciada uma sessão SSL entre o Profissional e o Portal da RTS, sendo assim concluido o processo de autenticação.

Capítulo 7

Avaliação

Neste capítulo é efectuada a avaliação das funcionalidades implementadas. O objectivo principal era pegar numa infra-estrutura de chave pública existente e adaptá-la ao uso do cartão de cidadão.

7.1 Arquitectura Avaliada

O Cartão de Cidadão não foi concebido para ser usado na arquitectura existente, por isso tiveram que ser feitas algumas alterações. Essas alterações foram feitas em função dos recursos disponibilizados pelo Cartão de Cidadão.

Foram analisadas várias soluções para a maneira como as credenciais dos profissionais fossem armazenadas. A solução escolhida foi guardar um certificado no espaço do Bloco de Notas do Cartão de Cidadão usando o mesmo par de chaves usado pelo certificado de autenticação fornecido com o cartão. Foi também considerada a hipótese de guardar o certificado fora do cartão, com a informação para a sua localização guardado no seu Bloco de Notas. À partida, a primeira opção é suficiente, porque é possível criar um certificado apenas com informação necessária que caiba no espaço disponível do Bloco de Notas.

É usado apenas um certificado e não dois, como anteriormente. Esse certificado serve-se do par de chaves de autenticação existente no Cartão de Cidadão, e o seu pedido necessita da existência do Cartão de Cidadão no acto em que é executado. O pedido tem que ser assinado com a chave privada do Cartão de Cidadão, e para que a operação de assinatura seja concluida, o seu dono tem que introduzir o código PIN. Isto impede que alguem que não conheça o PIN e consiga obter a

posse de um cartão faça pedidos em nome de outro.

Como a chave privada está contida no cartão para o qual foi pedido o certificado, não é possível usar um certificado emitido para um cartão noutro cartão diferente. Seria possível copiar o certificado pois ele encontra-se numa zona pública, capaz de ser visualizada por qualquer individuo que tenha acesso físico ao cartão. A sua tentativa de utilização num cartão diferente iria falhar porque a chave privada não iria corresponder à chave pública contida no cartão.

O pedido de certificados é feito usando o OpenSSL, com algumas alterações no seu código, permitindo o uso da PKCS#11, para poder aceder ao cartão e assinar o pedido com a chave privada do certificado de autenticação contida no Cartão de Cidadão. Esse código do OpenSSL foi alterado especificamente para a emissão de certificados que reutilizam o par de chaves de autenticação existente no Cartão de Cidadão.

Para aceder ao Bloco de Notas e conseguir que o *browser* lesse o certificado adicionado ao cartão, foi necessário implementar um *wrapper*. Esse wrapper chama o módulo da PKCS#11 fornecido com o Cartão de Cidadão, e permite ao browser reconhecer o certificado contido no Bloco de Notas do Cartão de Cidadão. O *wrapper* foi apenas testado no *browser* Firefox, e concebido para satisfazer os pedidos efectuados por este *browser*.

A sua utilização pelo Internet Explorer está sujeita a testes e provavelmente iria implicar algumas alterações na sua implementação, mas nada de muito significativo. A implementação pode ser melhorada, e uniformizada para funcionar com ambos os *browsers* mais populares. Seria mais eficiente a concepção de um módulo que incluisse a PKCS#11 funcional para o Cartão de Cidadão e o acesso ao Bloco de Notas para ler o certificado extra sem depender de outros módulos, tornando-o numa ferramenta mais robusta.

Outra funcionalidade que foi referida mas não chegou a ser implementada foi a de ler certificados contidos em locais cujo o endereço electrónico estaria guardado no Bloco de Notas. Isso iria permitir a leitura de vários certificados e não os limitaria em tamanho.

Capítulo 8

Conclusões

Neste documento foi apresentada uma adaptação possível de uma arquitectura de autenticação já existente, em que eram usados *smart cards* genéricos, ao uso do Cartão de Cidadão como meio de autenticar Profissionais na RTS.

A solução escolhida simplifica em alguns aspectos a arquitectura previamente definida, como por exemplo o uso de apenas um certificado e a eliminação de certificados cruzados, bastando que as Instituições de Saúde confiem na EC raiz da RTS e que a RTS confie nas EC emissoras das Instituições de Saúde.

Para os Profissionais, existe a vantagem de poder usar um cartão que já são obrigados a ter em sua posse. Para a RTS, tem a vantagem de não requerer dispositivos extra que poderiam sair caros à organização.

Como o Cartão de Cidadão é igual para todos, não há o problema que existia com os *smart cards* nas questões de incompatibilidades. O *wrapper* funciona com qualquer Cartão de Cidadão uniformemente, e o mesmo acontece com a criação de pedidos de certificados com o OpenSSL.

A gestão dos Profissionais continua a ser feita com *Active Directory* sem sofrer alterações, podendo manter a mesma estrutura já antes definida.

Este método de criação de certificados com extensões aproveitando as chaves contidas no Cartão de Cidadão pode ter várias aplicações, não só a RTS, mas qualquer tipo de infra-estrutura de chaves públicas que obrigue à existência de certificados com extensões.

Bibliografia

[1] Cunha, J.P.S., Cruz, I., Oliveira, I., Pereira, A.S., Costa, C.T., Oliveira, A.M., Pereira, A.: The RTS Project: Promoting secure and effective clinical telematic communication within the Aveiro region. In: eHealth 2006 High Level Conf., Malaga, Spain (2006)

[2] Gomes, H.: Autenticação em Sistemas Telemáticos Biomédicos. U.A. (2007)

[3] AMA: Manual técnico do middleware cartão de cidadão (June 2007)

[4] PKCS#11: Cryptographic Token Interface Standard, v2.20. RSALaboratories (2004)

[5] Helder Gomes, André Zúquete, J.P.S.C.: Arquitectura de autenticação baseada em certificados para a rede telemática de saúde (rts)

[6] André Zúquete, Helder Gomes, J.P.S.C.: Authentication architecture for region-wide e-health system with smartcards and a pki

[7] AMA: Autenticação com o cartão de cidadão (December 2008)

[8] UCMA/UMIC/DGRN: O novo documento de identificação dos cidadãos portugueses - nota informativa 1 (March 2007)

[9] SEMA/UMIC/AMA: Especificações leitores base (June 2007)

[10] Ali Sunyaev, Alexander Kaletsch, C.M., Krcmar, H.: Security analysis of the german electronic health card's peripheral parts (2009)

[11] André Zúquete, Helder Gomes, J.P.S.C.: Athentication of professionals in the rts e-health system (2008)

[12] Housley, R., Ford, W., Polk, W., Solo, D.: Internet X.509 Public Key Infrastructure Certificate and CRL Profile. RFC 2459, IETF (January 1999)

[13] Housley, R., Polk, W., Ford, W., Solo, D.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3280, IETF (April 2002)